University of Bath

PHD

Intelligent autoreclosing for systems of high penetration of wind generation with real time modelling, development and deployment

Le Blond, Simon

Award date:
2011

Awarding institution:
University of Bath

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 23. May. 2019

INTELLIGENT AUTORECLOSING FOR SYSTEMS WITH HIGH PENETRATION OF WIND GENERATION WITH REAL TIME MODELLING, DEVELOPMENT AND DEPLOYMENT

Simon Le Blond

A thesis submitted for the degree of Doctor of Philosophy
University of Bath
Department of Electronic and Electrical Engineering
March 2011

COPYRIGHT

Attention is drawn to the fact that copyright of this thesis rests with its author. A copy of this thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with the author and they must not copy it or use material from it except as permitted by law or with the consent of the author.

This thesis may be made available for consultation within the University Library and may be photocopied or lent to other libraries for the purposes of consultation.

Abstract

This thesis presents investigations into the effect of modern wind farms on grid side short circuits using extensive real time digital simulation. Particular reference is made to adaptive autoreclosing algorithms using artificial neural networks. A section of 132kV transmission grid in Scotland, including DFIG wind farms, is modelled on a real time digital simulator. An algorithm is then developed and tested using this model to show that this autoreclosing technique is feasible in systems with high penetration of wind generation. Although based on an existing technique, an important innovation is the use of two neural networks for the separate tasks of arc presence and extinction. The thesis also describes a low-cost, real time, relay development platform.

Executive summary of key achievements

- The effect of wind turbines on transmission line short circuit transients, with a comparison of the other significant parameters

- Treatment of unbalanced faults and realistic arc modelling in this context

- Feasibility studies on RTDS development of AdTAR using primary arcing and inter-circuit coupling

- Development of robust AdSPAR autoreclosing algorithm using twin neural networks

- A critical discussion of the use of AI in power system protection

- A low cost, IEC 61850 compliant, real time relay development platform

Acknowledgement

First and foremost, I would like to thank my supervisor, Professor Raj Aggarwal, for the patience and support he has shown me, and especially for taking the risk of agreeing to supervise a physics graduate in the first place.

Thanks to Mr Brian Ross for his technical advice and the huge effort of wiring the enclosure, and Mr David Parker for building it. In the power group, I found discussions with Dr Miles Redfern, Dr Rod Dunn, Dr Furong Li, James Brooks, Rob Dowley, Dr Vandad Hamidi, Hamza Alsafih, Hazem Zubi and Anthony Gee to be very useful. Dr Myo Aung was an enthusiastic and helpful colleague and his early passing is very sad. Thanks to all my other colleagues for their friendship and kindness.

At the University of Strathclyde, I would like to thank Ibrahim Abdulhadi and Professor Graeme Burt for collaborating within this work stream. Robert Macdonald and Dr Ryan Tumilty were kind enough on separate occasions to show me how Glaswegians celebrate the conclusion of a conference.

Thanks also to Dr Ray Zhang for his insight and warm-hearted supervision during my industrial placement, Paul Forsyth and his colleges at RTDS technologies for providing excellent and prompt advice from across the Atlantic, and Jim Henderson at Innovative Integration, Tim Bigg, John Owen and Kathy Ollington at Entegra for their help on the X3-SD.

I would like to thank the EPSRC and members of the Supergen:FlexNet consortium for their sponsorship. Thanks to all the academics and administrators of FlexNet for their hard work in running the courses and events: I found them useful, informative and enjoyable in equal measure.

On a more personal note, thanks to all my family and friends who have supported me throughout this PhD, especially through a period of illness. Thanks to Dad for

showing me what you know is important, and thanks to Mum for showing me what you don't know is just as important. Love to Sarah.

S. Le Blond

Bath, March 2011

Table of Contents

Abstract…….ii

Thesis Achievements……ii

Acknowledgment……...iii

Table of Contents……v

Glossary of terms……ix

List of figures……x

List of symbols……xv

List of tables ……xvii

Introduction ……1

I-1 Context
I-2 Power system protection
I-3 Autoreclosers
I-4 This thesis

Chapter 1 – Basic principles of protection……..5

1-1 Relays
1-2 History
1-3 Protection principles
1-4 Types of faults
1-5 Autoreclosing

Chapter 2 – Literature review on adaptive autoreclosure…….24

2-1 Introduction
2-2 History
2-3 Autoreclosure considerations
2-4 Early work on adaptive autoreclosing
2-5 Diagnosis of transient and permanent faults
2-6 Adaptive autoreclosure for single pole circuit breakers
2-7 Three phase adaptive autoreclosing
2-8 Optimal autoreclosure
2-9 Adaptive autoreclosing and power electronics

Chapter 3 – The effect of wind generation on transmission line short circuits…38

3-1 Introduction
3-2 Wind turbine types
3-3 Discrete wavelet transform
3-4 Study Method
3-5 Waveform indexing
3-6 Results
3-7 Conclusion

Chapter 4 – A real time model ……74

4-1 Introduction
4-2 RTDS background
4-3 RTDS hardware
4-4 132kV network
4-5 Conclusion

Chapter 5 – AI techniques in power systems…….103

5-1 Introduction

5-2 Artificial intelligence in power systems
5-3 Evolutionary computing and genetic algorithms
5-4 Multiagent systems
5-5 Expert systems and fuzzy logic
5-6 Artificial neural networks
5-7 Hybridised AI systems
5-8 Conclusion

Chapter 6 – Algorithm Development……123

6-1 Introduction
6-2 Autoreclosing philosophy
6-3 Primary arc
6-4 Harmonic emissions
6-5 Transient stability of 132kVsystem
6-6 Neural network separation
6-7 ANN training cases
6-8 Downsampling
6-9 Feature extraction
6-10 Normalisation
6-11 Architecture of neural networks
6-12 Time domain response
6-13 Conclusion

Chapter 7 – Real time digital simulation……...150

7-1 Introduction
7-2 Hardware overview
7-3 Software
7-4 Computational load assessment
7-5 Algorithm testing
7-6 Conclusion

Conclusion…….177

C-1 Introduction
C-2 Thesis summary
C-3 Discussion
C-4 Further work
C-5 Closing thoughts

References…….187

Appendix 1……194

1) RSCAD draft system model
2) RSCAD runtime

Appendix 2 ……...205

1) Algorithm C++ code

Related Publications…….247

Glossary of abbreviations

AA	**A**daptive **A**utoreclosing
AdSPAR	**Ad**aptive, **S**ingle-**P**ole **A**uto **R**eclosing
AdTAR	**A**daptive, **T**hree-**P**hase **A**uto **R**eclosing
AI	**A**rtificial **I**ntelligence
ANN	**A**rtificial **N**eural **N**etwork
ATP	**A**lternative **T**ransients **P**rogram
CVT	**C**apacitor **V**oltage **T**ransformer
DAR	**D**elayed **A**uto**r**eclosure
DFIG	**D**oubly **F**ed **I**nduction **G**enerator, a wind turbine with a partial frequency converter
DWT	**D**iscrete **W**avelet **T**ransform
DSP	**D**igital **S**ignal **P**rocessing
EMTP	**E**lectro**m**agnetic **T**ransients **P**rogram
EC	**E**volutionary **C**omputing
FC	**F**ull **C**onverter, a wind turbine with a fully rated frequency converter
FPGA	**F**ield **P**rogrammable **G**ate **A**rray
FACTS	**F**lexible **A**lternating **C**urrent **T**ransmission **S**ystem
FLOP	**Flo**ating Point O**p**eration
FLOPS	**Flo**ating Point O**p**erations Per **S**econd
GA	**G**enetic **A**lgorithm
IED	**I**ntelligent **E**lectronic **D**evice
IGBT	**I**ntegrated **G**ate **B**ipolar **T**ransistor
LCS	**L**earning **C**lassifier **S**ystem
MAS	**M**ulti**a**gent **s**ystem
PSO	**P**article **S**warm **O**ptimisation
RTDS	**R**eal **T**ime **D**igital **S**imulator
THD	**T**otal **H**armonic **D**istortion
SA	**S**imulated **A**nnealing
SPG	**S**ingle **P**hase to **G**round (fault)

VSC **V**oltage **S**ource **C**onverter

List of figures

Chapter 1

1.1 X/R characteristic for distance relay

1.2 Figure 1.2 X/R characteristic for mho relay

1.3 Symmetrical components

1.4 Single phase to ground fault

1.5 Phase to phase fault

1.6 Double phase to ground fault

1.7 Three phase fault

1.8 Three phase to ground.

1.9 Lightning as current source

1.10 Autoreclose timeline for transient fault

1.11 Autoreclose timeline for permanent fault

1.12 Minimum arc deionisation time

Chapter 2

2.1 Variance of sending end voltage with time fixed resistance

2.2 Variance of sending end voltage with time realistic arc model

2.3 Fitton et al method of AdSPAR

Chapter 3

3.1 Schematic of DFIG Design reproduced from

3.2 Schematic of Full Converter Design reproduced from

3.3 Back to back frequency converter

3.4 Discrete Wavelet Transform algorithm sampling

3.5 UKGDN, EVH-2

3.6 Test network

3.1.1 Faulted waveform for conventional default case

3.1.2 Faulted waveform for full converter default case

3.1.3 Faulted waveform for DFIG default case

3.1.4 Wavelet transform details for full converter default case

3.1.5 Wavelet transform details for conventional default case
3.1.6 Wavelet transform details for DFIG default case
3.2.1 Faulted phase waveform for conventional 150 MW case showing double transient at 340ms
3.3.1 Wavelet transform details for full converter 35 GVA case
3.3.2 Wavelet transform details for DFIG 35 GVA case
3.4.1 Conventional generation, 50 km line
3.4.2 The full converter case, 50 km line
3.4.3 The conventional case, 5 km line
3.4.4 The full converter case, 5 km line
3.4.5 The DFIG case, 5 km line
3.5.1 Wavelet transform for the conventional case for three phase to ground fault
3.5.2 Wavelet transform for the DFIG case for three phase to ground fault
3.6.1 Full converter case 100m from sending end (measuring bus)
3.6.2 Full converter case 100m from receiving end
3.6.3 Phase B, Full inverter 100 from sending end
3.7.1 The full converter case for voltage zero fault inception
3.7.2 Phase B the full converter case at for voltage zero fault inception
3.8.1 Phase B for the Full inverter at 50 Ω fault resistance
3.8.2 Healthy phase for the FC at 0 Ω
3.8.3 Wavelet transform details for healthy phase of of the FC at 50 Ω
3.8.4 Wavelet transform details for healthy phase of the Full inverter at 0 Ω
3.10 The B phase over 1 second, FC default case
3.10.2 The B phase over 1 second, DFIG default case

Chapter 4

4.1 A resistor as a time dependent circuit element
4.2 A capacitor as a time dependent circuit element
4.3 Trapezoidal rule of integration
4.4 An inductor as a time dependent circuit element
4.5 Equivalent circuit for a two terminal line with distributed parameters
4.6 Map of network modelled

4.7 Diagram of network modelled
4.8 Line type 1
4.9 Line type 2
4.10 Line type 3
4.11 The high fidelity area of network modelled
4.12 The high fidelity section of network modelled
4.13 The DFIG model in RSCAD draft showing the machine, VSC, filter bank and step up transformer
4.14 Equivalent circuit diagram of CVT
4.15 Fault and breaker control

Chapter 5

5.1 Research trends in AI: percentage of IEEE papers containing these AI techniques in the title or key word section
5.2 A fuzzy membership function: the day's quality dependence on rain
5.3 A fuzzy membership function, the day's quality dependence on sunshine
5.4 The neuron model
5.5 Common transfer functions
5.6 MLP architecture

Chapter 6

6.1 Primary arc detection
6.2 Transient and permanent post-fault, pre breaker waveforms
6.3 Frequency spectra of primary arc and permanent resistance
6.4 Spectrogram plot of transient fault no harmonics or CVT
6.5 Spectrogram plot of permanent fault no harmonics or CVT
6.6 Spectrogram plot of transient fault with CVT
6.7 Spectrogram plot of permanent fault with CVT
6.8 Spectrogram plot of transient fault with harmonics
6.9 Spectrogram plot of permanent fault with harmonics
6.10 Spectrogram plot of transient fault with harmonics and CVT
6.11 Spectrogram plot of permanent fault with harmonics and CVT
6.12 Reverse spectrogram plot of transient

6.13 Transient fault with harmonics and CVT

6.13 Permanent fault with harmonics and CVT
6.14 Two machine system
6.15 Power transfer dependence on angle difference
6.16 Transmission angle against single phase transient fault
6.17 The distinction between safe and not safe to reclose
6.18 The algorithm visualised in Simulink blocks
6.19 Hann window function
6.20 Time domain ANN responses to transient fault
6.21 Time domain ANN response of permanent fault.

Chapter 7

7.1 Real time development platform
7.2 Interface module enclosure design
7.3 Interface module circuitry
7.4 Interface module circuitry
7.5 SNAP program configuration tab
7.6 SNAP program setup tab
7.7 SNAP program configuration tab
7.8 SNAP program configuration tab
7.9 Real time response of CVT to transient fault
7.10 Farr bus response to transient fault
7.11 Current in Farr line sections, transient fault
7.12 Farr bus response to permanent fault
7.13 CVT response to permanent fault
7.14 A failed reclose onto permanent fault, Farr bus
7.15 A failed reclose onto transient fault before arc extinguish, Farr bus
7.16 Line current response to permanent fault
7.17 CVT response to transient fault, no arc extinguish
7.18 Ferro-resonance in CVT
7.19 Secondary arcing on A phase and mutual inter-circuit coupling on other phases
7.20 Typical secondary arcing on A phase and due to inter-circuit coupling

7.21 Close up typical secondary arcing on A phase due to inter-circuit coupling
7.22 Permanent fault with three-phase breaking, note voltage collapse on faulted phase
7.23 Close up permanent fault with three-phase breaking

List of symbols

In order of appearance:

Symbol	Description
$\underline{Z}$	impedance
$\underline{I}$	complex current phasor, A
$\underline{V}$	complex voltage phasor, V
α	unit operator, $\angle$ 120 °, –0.5 + 0.866j
μs	microsecond, 1 x 10^{-6}s
a_0	mother wavelet scaling term
b_0	mother wavelet scaling term
Fs	sampling frequency, in Hz
Δt	simulation timestep, s
v_m	instantaneous voltage value at node m, V
v_k	instantaneous voltage value at node k, A
i_{km}	branch current between nodes k and m, A
$I_{k,m}(t-\Delta t)$	history value of current source in the previous timestep, A
τ	Travel time from node k to m in distributed parameter line, s
L'	Inductance per unit length, H.m^{-1}
C'	Capacitance per unit length, F.m^{-1}
u	Phase velocity, m.s^{-1}
Z	Surge impedance, Ω
[Y]	Admittance matrix
[v(t)]	Instantaneous nodal voltage matrix
[i(t)]	Instantaneous nodal voltage matrix
[I]	Known current source matrix
S.C.C.	Short circuit capacity, *GVA*
C_1	CVT capacitor divider, system side capacitance

C_2	CVT capacitor divider, grounded side capacitance
ω_0	System angular frequency, rad.s^{-1}
g	Time dependent arc conductance, Ω^{-1}
G	Stationary arc conductance, Ω^{-1}
l_p	Primary arc length (distance between tower arcing horns), m
U_p	Rate of rise of primary arc voltage, V.m^{-1}
τ_a	Arc time constant, s
τ_s	Secondary arc time constant, s
V_r	Secondary arc voltage restriking value, V
l_p	Primary arc length, m
l_s	Secondary arc length, m
U_s	Rate of rise of secondary arc voltage, V.m^{-1}
w_{kp}	Weighting coefficient to neuronal input
θ_k	Neuronal bias
δ	Angular difference between two generators, degrees
Y_{out}	Normalised output at frequency band
$X_{\max}$	Maximum average value of signal at frequency band, V
$X_{\min}$	Maximum average value of signal at frequency band, V

List of tables

Table 3.1	Line parameters for branch 106 - 107
Table 4.1	Model source parameters
Table 4.2	Load parameters
Table 5.1	Summary of autoreclosing task and the most suited AI solution
Table 6.1	G 5/4 recommendations for the first 9 harmonics on systems between 20kV and 145kV
Table 7.1	Real time testing of algorithm

I - Introduction

I-1 Context

Today the security of supply faces a multitude of threats unlike it has ever known. Fossil fuel based generation is becoming far less attractive due to the damaging emission of CO_2 and other greenhouse gases. At time of writing, the consequences are beginning to slide beyond environmental into real economic terms. Moreover, with an unstable geopolitical situation, for many nations, the imported supply of fossil fuels is not particularly secure, especially considering sources are finite and diminishing at an increasing pace. These pressures did not exist half a century ago and thus electrical power systems were designed with centralised, fossil fuel based generation in mind. A consequence of this in the UK and other developed nations, is that much of the grid infrastructure is over 60 years old and is rapidly approaching the end of its lifespan. The debate over nuclear power is very difficult to conclusively resolve and in recent years, a lack of decisive policy over the issue has hampered progress. The building and development of nuclear plant can be upwards of ten years for individual stations and currently the UK is decommissioning more old plants than it is building. It is well known that nuclear fission is a mature yet expensive technology. It has had a chequered history, particularly since nuclear weapons were developed during the close of WWII. The waste products require safe storage for thousands of years before radioactivity decays to acceptable levels. Who can guarantee that future societies can be responsible or stable enough to be the gatekeepers of this unpleasant legacy? On the other hand, fusion, would be an ideal energy solution for the opposite reasons that fission is not. All being well, the first critical yield fusion reactor (ITER) will come online in 2016, and generate a positive yield in 2025 [1]. However, fusion remains in experimental stages: it is a long way off the commercial investment cycle and as it remains critically underfunded, will only be available 30-50 years in the future.

What will fulfil the generation needs in the near to mid term? There is no one magic bullet for such a complex problem; the only feasible solution is a multifaceted one.

Only a comprehensive strategy will diffuse problems with renewables such as intermittency and cost-effectiveness. There must be ongoing investment in micro-generation, wind, solar, tidal, wave, hydro, combined heat and power, energy storage, carbon capture and storage, sustainably sourced biofuels, low carbon transport, and accompanying network infrastructure and technology. The benefits of growth in these industries can soon become economic as well as environmental. The UK has the potential to become leaders in the field, exporting expertise and technology worldwide.

The latest government target is 20% of the supply by renewable energy in 2020 [2] Given the scale of the impending crisis, this perhaps seems a little tentative. But as far as power system planning and design is concerned, such a goal demands a paradigm shift.

Despite the ambiguity regarding nuclear power, it is widely accepted that future networks will need to evolve to cope with growing renewable generation. This will be connected increasingly at distribution levels, and in areas of low population density. Therefore, networks will require bi-directional power flows over longer distances. Today power flows mainly from large conventional plants through the transmission network down to customers connected at distribution level. In future, network topologies will be far more active, their configuration at all levels dynamic due to the intermittent nature of renewables. Coupled to this, the increased demand for power, and the social and economic inertia involved in building new infrastructure will require existing networks to be driven harder. Transient stability margins will decrease as operators push more power through the networks. Fault clearance times will need to improve to preserve stability, prevent asset damage and maintain the security of the supply.

In order to meet these challenges, future power grids must become smart and flexible. The transistor ushered in the information age, bringing tremendous advances in electronics - notably telecommunications, computer science and microprocessor technology. These must now be fully applied in power systems so that the life-force underpinning all our technology is secured for generations to come.

I-2 Power system protection

Power system protection is vital for safety, maintaining security and quality of supply and minimising equipment damage. It also has a key role to play in the smart grid revolution. Protective relays must now evolve to be adaptive, continuously updating settings to protect a system whose topology is constantly changing.

Automatic protection devices are usually the first line of defence against faults. Of these, unit protection is specialised for a single unit of power system equipment, whereas non-unit serves to protect a zone. The difficult compromise between security and dependability must be reached so that relays only trip for faults occurring inside their protective zone or item of equipment. An oversensitive relay will lead to sympathy trips in response to abnormalities originating from beyond the protected zone. If automatic protective devices fail to deal with problems, human intervention is required to isolate the faulted equipment and preserve the operation of the rest of the power system. The timescales on which corrective control actions can be manually implemented are considerably longer (seconds and minutes), and by then the fault may cause adverse consequences elsewhere. In comparison, with automatic protective devices, the correct action can potentially be taken within a few power frequency cycles (ms).

Of all faults, approximately 60% occur on overhead lines [3]. These are the most exposed area of the power system and thus vulnerable to lightning, wind, snow, sleet, rain, fog, vegetation, birds and animals. The majority of faults on transmission lines are transient short circuit faults, usually caused by a lightning strike on or near the tower or line.

I-3 Autoreclosers

A circuit breaker is the vital component that isolates the faulted equipment from the rest of the power system. Its operation is governed by the protective relay relevant to the zone or equipment to which it is assigned. Autoreclosers are circuit breakers that

are able to re-energize shortly after they have been tripped, and thus take advantage of the transient nature of most faults. They are often programmed to attempt a number of reclosure events, and if the fault fails to clear, lock out permanently.

The disadvantage of conventional autoreclosers is the possibility of unsuccessful reclose attempts, in the case of permanent or long duration transient faults. An unsuccessful reclose attempt causes more wear on expensive equipment, such as generator drive shafts or by causing local generator oscillations, and has negative implications for system stability and maintaining synchronism. Reclosing onto a transient fault too early can lead to the fault re-establishing itself due to the products of the arcing path failing to clear. These problems can be overcome by using the fault signature to issue a reclose decision at the optimum moment, should it be appropriate at all. Thus, a single optimal reclosure decision may be issued for a transient fault and reclosure blocked for permanent faults. This advanced form of autoreclosure is known as adaptive autoreclosure.

I-4 This thesis

In the short to mid term, a vast majority of UK renewables will come from wind generation. In 2010, the UK passed 5 GW mark of installed capacity, 3.7 GW of which was onshore [3]. This thesis investigates adaptive autoreclosing on systems with high penetration of wind farms. A particular emphasis is made on the power electronic converters signatures and their impact on the grid. The findings are therefore also pertinent to other forms of asynchronous generation and storage where frequency conversion is required. In order to substantiate the findings, an Artificially Intelligent algorithm is developed, tested and demonstrated in real time using a Real Time Digital Simulator.

Chapter 1 - Basic Principles of Protection

1-1 Relays

In the context of HV power, a relay is a basic computational unit of secondary system equipment. It monitors quantities at one or several points in the system, and in the event of a fault, issues the trip signal to the circuit breakers. In the transmission system, the main function of protection is to de-energise the faulted part of the system as quickly as possible, preserving the stability of the system and preventing the loss of synchronism. A secondary outcome is to minimise damage to the equipment experiencing the fault. Speed of operation is paramount to both these considerations. Fault clearing times determine how much power can be transferred through the system without loss of stability, should a fault occur. The energy liberated at the point of the fault is proportional to the square of the fault current times the duration of the fault, so that the longer the fault lasts, the more damage occurs. At transmission level, system wide stability and security is essential. At distribution level, the main function of protection is to maintain supply to as many customers as possible whilst isolating and minimising damage to faulty equipment.

Careful consideration must be taken in coordination of protection schemes. This is usually done on a zone by zone basis, with each scheme monitoring a protected zone. The zones overlap such that no area of the power system is left unprotected. Due care must be placed so that for any given fault condition, the right protection operates and does not cause sympathy tripping by other relays. The zone covered by a relay is also known as the reach. A zone may protect one component or several. Coordination of the protective relays is therefore a broad and complex subject in its own right. In this chapter, only the basics will be discussed to put this thesis in context.

It is worth considering a protection scheme is further limited by the response time of the mechanical circuit breaker that the relay controls. The design of circuit breakers and switchgear is a complex sub-branch of electrical and mechanical engineering. Circuit breakers are normally only designed to operate around a current zero and even then must have good arc quenching capability. Arc quenching mediums are high pressure air, oil or the increasingly common gas, sulfur hexafluoride known as SF_6.

1-2 History

Over the last century, advances in electronics have been reflected in evolving relay technology. The first generation of relays were based on electromechanical principles; the relay contacts were physically operated, usually by a threshold current caused by a fault condition. The required level of energisation would result in sufficient induction or magnetisation of some moving part, resulting in mechanical movement such that contacts were made or broken. Electromechanical relays performed reliably for over fifty years and are still used today in some simple protection applications. Solid state technology, particularly the invention of the transistor made the static relay possible. The term static is used since it uses no moving parts – instead, it uses analogue circuitry to process transducer signals. The circuitry was fixed to perform a particular protection function, so several physical units where required at a substation to detect various fault scenarios. The decreasing cost of microprocessors ushered in the digital relay, whose use became widespread in the 1980s. In the digital relay, the analogue transducer signals are digitally sampled by analogue to digital converter. These signals are processed in a microprocessor. This offers the advantages of greater user definability. A subtle improvement to this design is the numeric relay. The numeric relay builds on its predecessor, with a dimensional jump in performance, using several onboard processors optimised for digital signal processing (DSP). Complex protection algorithms are designed in the software and evaluated in real time by the onboard processor(s). The increase in processing ability allows many user-definable protection functions to be provided by a single unit. A numeric relay will often have remote link access so that settings may be updated from the control centre [4]. In modern substation nomenclature a numeric relay is a type of intelligent electronic device, a term often abbreviated to IED. An important recent development is the IEC standard 61850, a modern standard for the automation of substations, with particular reference to the digital communications architecture [5].

1-3 Protection principles

Although the technology used in relay manufacture may have changed, the underlying operating principles of protection equipment have remained the same. The relay must respond to a change in the system quantities that indicates a fault condition. Real

world events can often be complex and cause faults to evolve, becoming more complex with time. Protection must operate reliably, therefore the field is conservative with equipment based on simple, elegant design.

Overcurrent protection

The simplest of all protection is overcurrent, which is of course the operating principle behind the humble domestic fuse. Short circuit fault conditions usually lead to abnormally high currents due to the reduced resistance. Careful consideration has to be made to ascertain what constitutes a fault current within the protected zone. Different overcurrent schemes are normally calibrated on a graded current or time basis, or ideally a combination of both, so that the relay closest to the fault trips first. If this fails to trip then the neighbouring relay will operate, followed by its neighbours, and so on, in a radial direction outwards from the fault location [6].

Differential protection

Differential protection monitors the difference in current entering and leaving a protected zone. For through faults, i.e. faults outside the protected zone, there will be no difference in the current at each zone boundary. A fault inside the protected zone will cause an instantaneous overcurrent. The difference in impedances from the fault point to each of the zone boundaries will result in different currents flowing at the zone boundaries, indicating a fault condition. This method requires transducers at each end of the protected zone and a secure communication channel between them. As such, it is most suited to unit schemes covering a single localised piece of equipment such as a generator, busbar or transformer [4].

Distance Protection

Distance protection operates on the impedance seen by a relay. The distance from the relay to the fault is proportional to the impedance seen by the relay. In the event of a fault, the impedance is reduced. The relay is calibrated such that it operates if the measured impedance lies inside the reach of the relay. Current and Voltage phasors are used to determine the impedance since according to ohm's law (1.1).

$$\underline{Z} = \frac{\underline{V}}{\underline{I}} \qquad (1.1)$$

The operating characteristic can be visualised on the X/R diagram, figure 1.1.

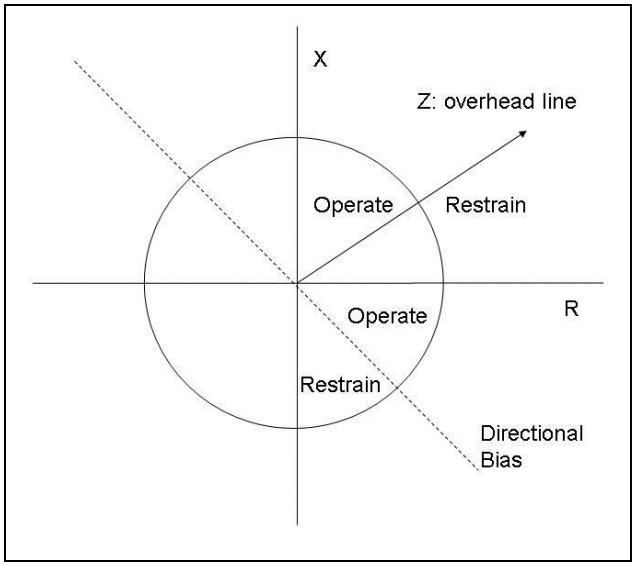


Figure 1.1 X/R characteristic for distance relay

The relay operates when it measures the line impedance to lie within the circular boundary. For pure impedance relays, a separate directional bias must also be included to discriminate in which direction the fault lies. In electromechanical relays, this can lead to 'contact races' between the directional element and the relay itself, causing maloperation. An improved concept operates on the reciprocal of Z, the electrical admittance; the convention is to refer to this as a "mho" relay after the SI unit of admittance. Using the pre-fault voltage as a polarizing signal gives the mho relay an inherent directional bias, so there is no need for a separate directional element. The mho relay X/R characteristic is shown in the figure 1.2.

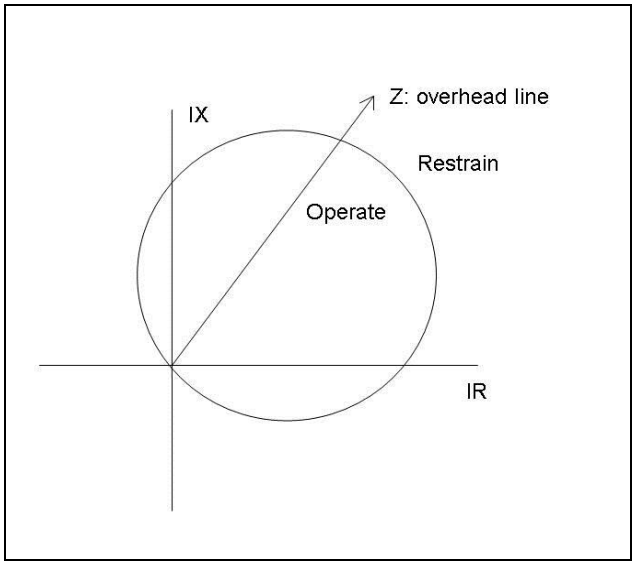


Figure 1.2 X/R characteristic for mho relay

Due to this property of bias, the mho relay is the most widely deployed distance relay. There are additional considerations that must be made for the extra resistive impedance within the fault path itself, either through an electric arc or vegetation. Also in the case of heavy loading on the line, the resistance of the line may falsely indicate a fault condition. This can be accounted for by changing the X/R characteristic so that it is elliptical rather than circular. The X/R charactersistic can be manipulated in other ways to achieve better selectivity.

The distance relay is vulnerable to power swings. These are sudden changes in network topology caused by loss of large loads or generation, network switching or line autoreclosure. They result in a sudden change in power flow, and a concomitant change in the voltage and currents. Extra logic may be needed to discriminate between power swings and fault conditions. (In some extreme unstable power swings, it may be desirable to split the power system into two balanced sections to prevent loss of synchronism but a distance relay alone is not capable of this action).

Transient based protection

The relay must respond to a change in system quantities that indicates a fault condition. The operating principles of relays originate from what was possible with early electromechanical designs. However, due to widespread and long term use, the advantages and disadvantages of these methods are now well understood. Therefore,

numeric relays have individual relay elements that reproduce the approaches used in traditional relay designs.

Conventional forms of protection all have one common principle; they operate by measuring quantities at the power frequency. When a fault occurs in a power system, there is a short transient period where the system experiences high frequency oscillations in response to the step change before it settles to a steady state [7]. In order to deal with pollution from other frequencies, relays must band-pass filter the signal so that only the power frequency signal is measured. System quantities, such as phase and magnitude, must be measured over at least two cycles. This places a lower limit on the time in which protection can operate - the operation time must be at least greater than one cycle. The transient period can also cause inaccuracies in the measurement of system quantities. Also traditional forms of protection are vulnerable to any phenomena occurring at the power frequency that could falsely indicate a fault condition and cause the relay trip spuriously or "maloperate". These include electromechanical oscillations caused by generators, low frequency resonance associated with reactive compensators, Ferro resonance or power swings caused by rapid changes in system topology, originating with faults or switching in and out of large loads.

Improvements in computer technology and time domain simulation of power systems have lead to the possibility of analysing the transient period and developing forms of protection based on this signal. Transient based protection makes use of more information contained in the wide band signal. This requires a high sampling rate, at least twice the highest frequency of interest due to the nyquist criterion. This is limited by the bandwidth of the transducers, and the bandwidth of the communication channels between them. They must be able to faithfully reproduce the system quantities up to the frequency used in the protection logic. This requires very fast digital signal processing by the relay to analyse this information in real time. In particular, frequency transforms are computationally expensive. However, improvements in digital signal processors and memory mean that modern numeric relays are capable of these requirements.

Development of transient protection requires very accurate time domain modelling of the primary system under both normal and fault conditions. Today, there are many power system transient programs available, which usually rely on the trapezoidal integration method, developed by Dommel and others, and first used in the Electrogmagnetic transients program (EMTP) [8]. Accurate modelling ensures that a trip decision is robust and able to discriminate between fault and non-fault conditions. Pollution can occur across the whole frequency spectrum; for example, harmonics caused by non-linear loads, power electronics, control equipment response or electromagnetic interference from nearby communication networks. It is essential to combine extensive simulation studies with real-world fault data to ascertain quantitatively how a transient based relay may be affected by some of these phenomena.

As a basis for improving power system protection, transient based protection offers two potential advantages; increased speed and immunity to power frequency phenomena. However, when both forms of protection are deployed in parallel there is a further increase in selectivity, and many of the potential problems associated with either are overcome. This approach requires the protection engineer have a deep knowledge of both methods. Moreover, access to reliable data will accurately simulate the systems transient response is essential.

Symmetrical Components for three phase systems

Of vital importance in protection of three phase systems is the tool of symmetrical components. This is a method by which the protection engineer can analyse unbalanced three phase faults with a set of balanced phasors. In 1913, Charles L. Fortescue showed that any multiphase system of *n* phases can be described by the equivalent *n* balanced systems [9].

So for a three phase system, the actual system current and voltage phasors can be expressed as three sets of three *balanced* phasors. These are designated the positive, negative and zero sequences, see figure 1.3. This greatly simplifies the analysis of unbalanced systems.

Mathematically, the relations between the phase quantities and the sequence components can be represented in matrix form:

$$\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & \alpha^2 & \alpha \\ 1 & \alpha & \alpha^2 \end{bmatrix} \begin{bmatrix} Va_0 \\ Va_1 \\ Va_2 \end{bmatrix} \qquad (1.2)$$

$$\begin{bmatrix} Va_0 \\ Va_1 \\ Va_2 \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & \alpha & \alpha^2 \\ 1 & \alpha^2 & \alpha \end{bmatrix} \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} \qquad (1.3)$$

Where α is the operator of unit magnitude and phase shift of 120 degrees, equation (1.4) and α^2 the operator of unit magnitude and phase shift 240 degrees, equation (1.5).

$$\alpha = \angle\, 120\,^\circ = -0.5 + 0.866\mathrm{j} \qquad (1.4)$$

$$\alpha^2 = \angle\, 240\,^\circ = -0.5 - 0.866\mathrm{j} \qquad (1.5)$$

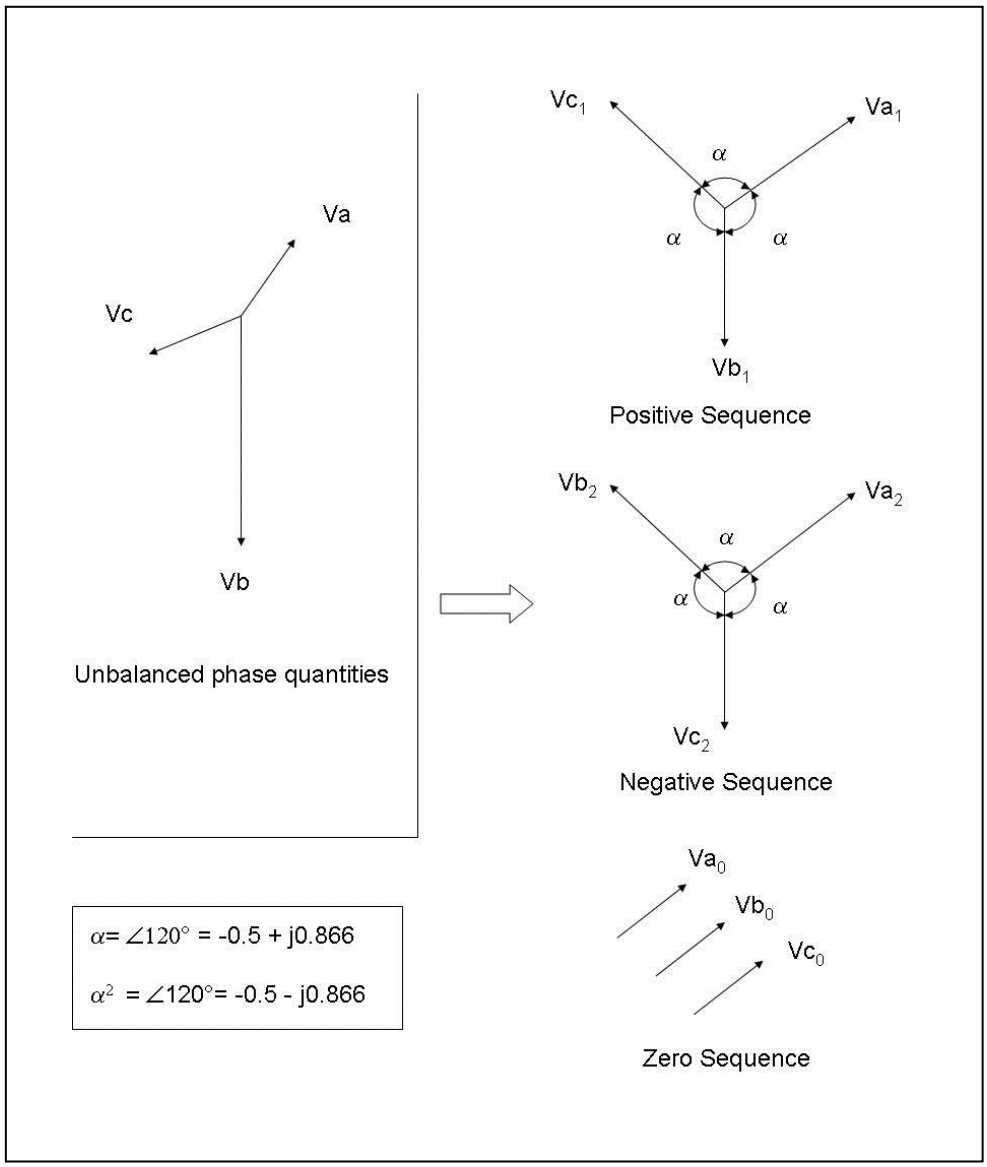


Figure 1.3. Symmetrical components

In balanced systems, the negative and zero phase sequence quantities are zero and the positive phase sequence is identical to the actual phase quantities. When the fault occurs, currents are set up in the zero and/or negative sequences. These phasors do not exist as physical quantities, but combine to form the unbalanced quantities. However, it is possible to evaluate them from the unbalanced system quantities such that they can be used to trip a relay. This can be done using circuitry or software that performs the transformations described by matrices (1.2) and (1.3). For example, in a three phase system, faults involving ground result in zero phase sequence current. The relay can be calibrated to operate on detection of zero phase sequence currents. This is useful for knowing the fault configuration, i.e. whether the fault is phase to phase or

phase to ground etc. Separate calibration settings may be desirable for different configurations.

In areas of balanced network, the sequences are independent (so for example, only negative phase sequence currents will produce negative phase sequence voltage drops). Hypothetical circuit diagrams for the three sequences may be visualised, to illustrate how the sequence networks will behave. In areas of unbalance, the sequence networks are no longer independent with current flowing in each, producing voltage drops in two or in some cases, all three sequence networks. Sequence relays are able to use this to discriminate between faulted and non-faulted areas.

It is important to bear in mind that these relationships only hold true for steady state systems at the fundamental system frequency. Careful consideration must be taken for the transient period following a fault. Filters must be used so that only the fundamental frequency is measured, but even so, this is an approximation until the higher frequency transients have attenuated [10]. All but transient based protection suffers from this limitation.

1-4 Types of faults

A large number of different fault types can occur in electrical systems. In a three phase circuit, one or several conductors may be shorted to each other or to ground in any number of combinations. The three phase circuit diagrams for these are shown in figures 1.4 – 1.8. (The diagrams assume the network voltage sources are wye-grounded connections, and are fed from both ends of the faulted zone. This would be true for a meshed transmission system where the faulted zone is an overhead line. In many circumstances these could be ungrounded wye or delta connections, and in radial networks the fault would only be fed from one end. Although the diagram is to illustrate the fault configuration only, it should be noted that the resulting unbalanced current is heavily dependant on the local network configuration).

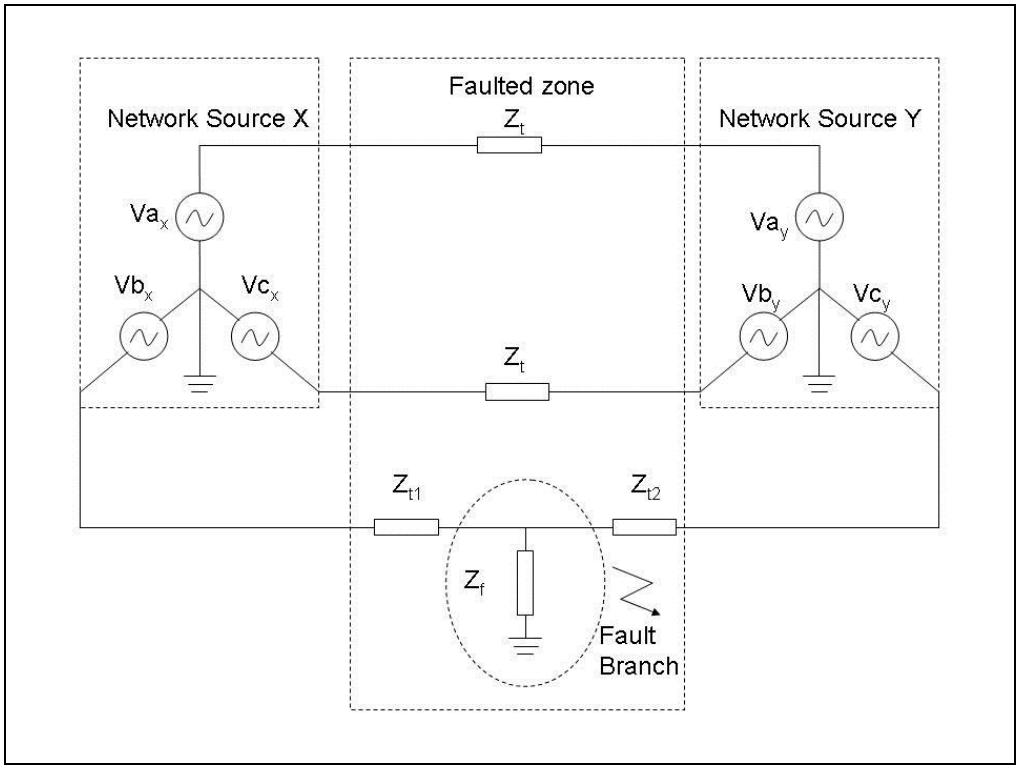


Figure 1.4 Single phase to ground fault

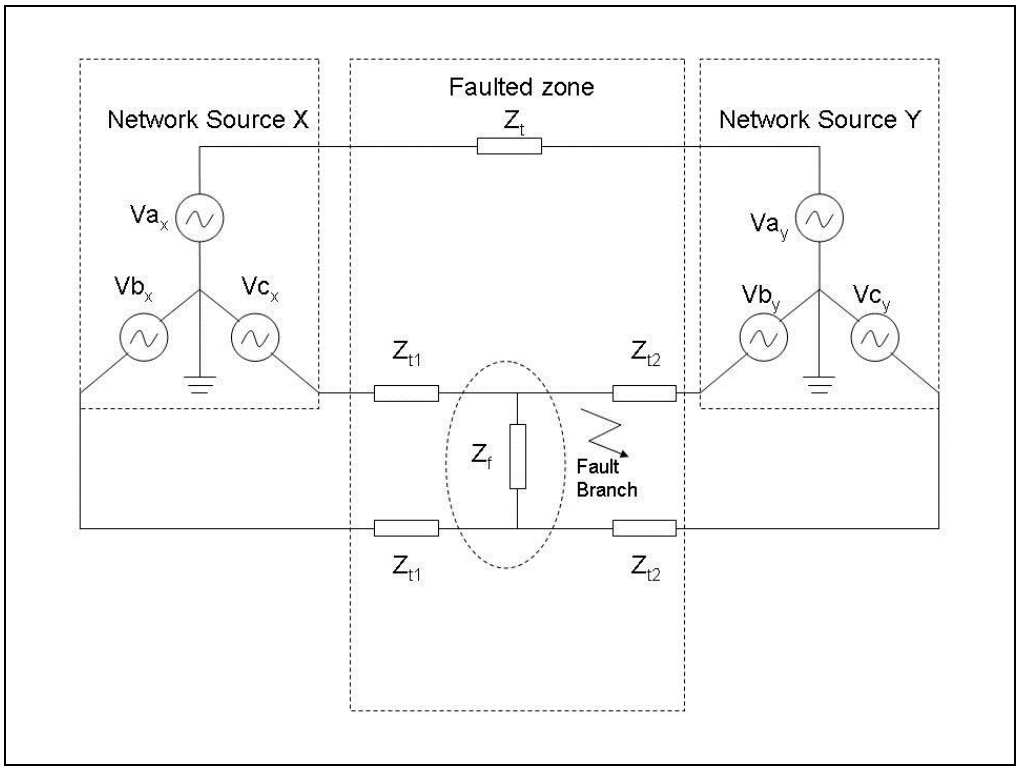


Figure 1.5 Phase to phase fault

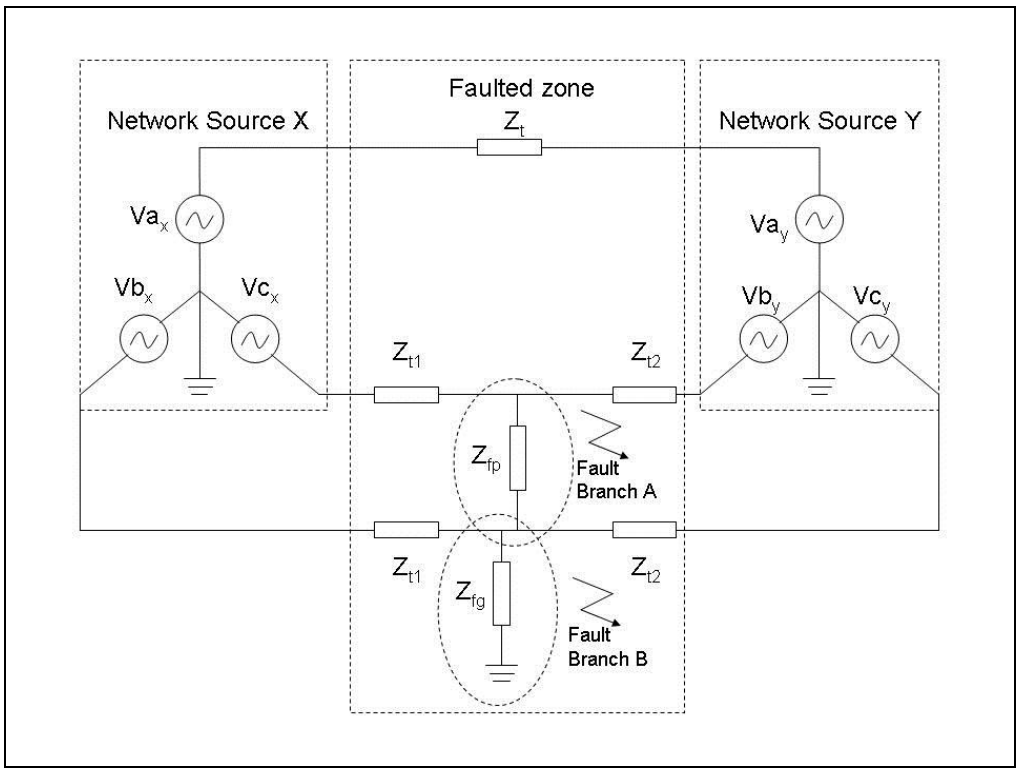


Figure 1.6 Double phase to ground fault

In the case of the double phase to ground fault there are two impedances. The phase to phase impedance (fault branch A) and the phase to ground (fault branch B). In figure 1.6, the phase to ground branch is on phase B but this may occur on either phase.

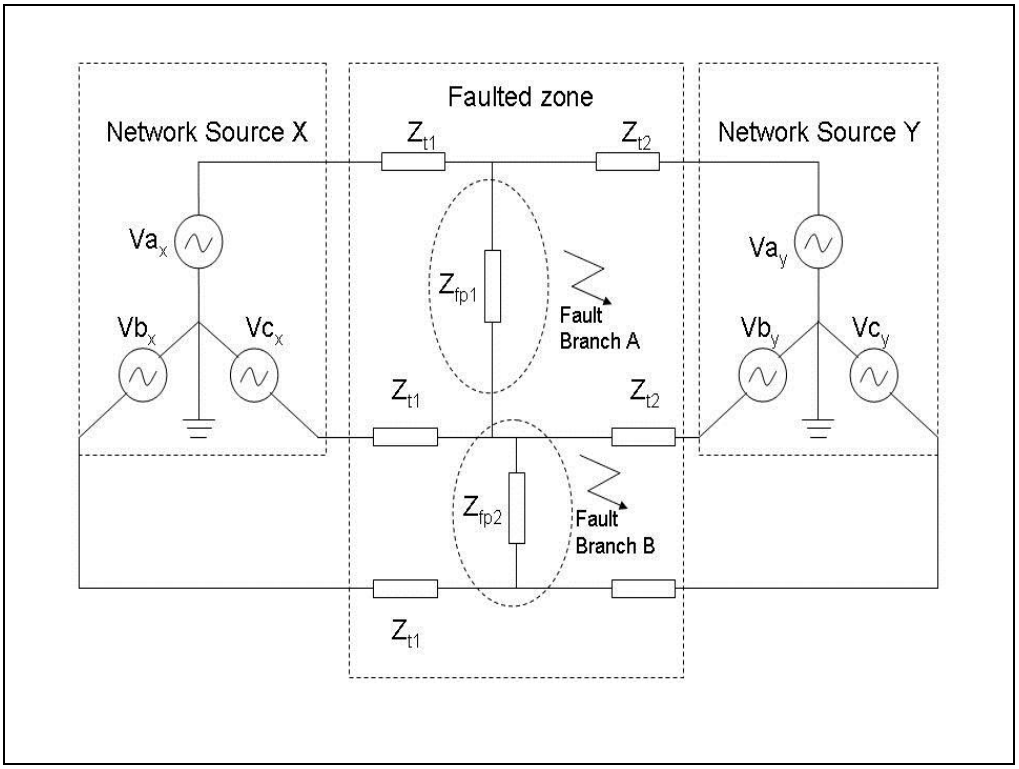


Figure 1.7 Three phase fault

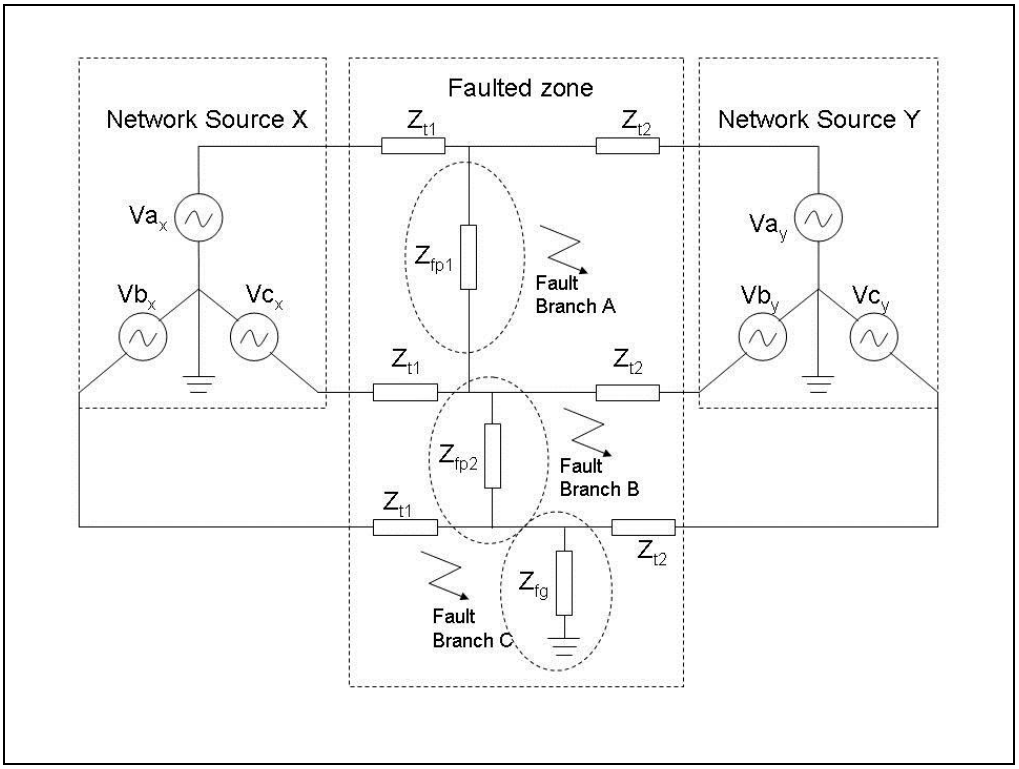


Figure 1.8 Three phase to ground

It is worth noting that it is possible to have a three phase fault involving three fault branches between the phases rather than two, but this is often neglected due its rarity and because the fault branch impedances are so low that they may be assumed to be a pure short circuit in most cases.

Indeed, a major assumption made in fault analysis is that each fault branch is often considered to be a pure short circuit with zero impedance, known as a solid fault. In reality, this is an approximation, especially if an arc is involved which has a dynamic resistance.

Modern specialist computer software can be used to obtain a more accurate system response. However, extreme caution must be used to obtain accurate input data in the right format and units. Without the right input data the results are of course meaningless. It is therefore vital that engineers continue to learn and employ symmetrical components to provide sanity checks on computer simulations.

Transient and permanent faults

There are many causes of faults in electrical power systems. These can lead to temporary (transient), semi-permanent or permanent faults. Most faults occur on overhead lines since they involve exposed, bare conductors.

Some of the causes of (overhead line) faults

- Weather – Predominantly lightning, but can also be caused by rain, wind, snow, ice and salt spray from the sea. This usually leads to transient arcing faults but in extreme conditions, strong winds and debris may cause a transmission tower to collapse or a conductor to break.
- Vegetation – In wooded areas vegetation may grow too near the conductors. In these cases the fault path to ground is often high impedance, and the fault is semi-permanent. The same fault may occur several times in quick succession as trees are repeatedly blown onto lines, and the vegetation may be burnt away over time. For the purposes of fast autoreclosing, a semi permanent fault can be treated as permanent.

- Animals – Animals may have the misfortune to short conductors, especially in distribution systems where the inter-conductor spacing is in the order of a large bird's wing span. In the past, animals wandering onto switching yards have had catastrophic consequences.
- Humans – The public can cause faults by activities such as angling, parasailing, hang gliding or kite flying. These kinds of faults are more likely in distribution systems. Technical crews working on live lines may inadvertently cause faults. This carries an extra safety dimension when considering re-energising the line, and prior knowledge of human proximity, utilities must disable autoreclosing relays.
- In areas of tectonic instability, earthquakes, volcanoes and tsunamis may cause faults.
- Aviation or motoring accidents are extremely rare but possible.
- Terrorist attacks, both physical or virtual, are also a possibility.

Lightning is responsible for around 80% of transmission line faults [11].
When lightning occurs, it usually does so as a series of short strokes between the cloud and ground. From the transmission line standpoint, only the heavy current return strokes are significant. Each of these strokes is equivalent to a short duration heavy current source. The current, which is normally between 10kA and 100kA, rises exponentially to a maximum in about 1-4μs and then attenuates to zero exponentially over approximately 50μs.

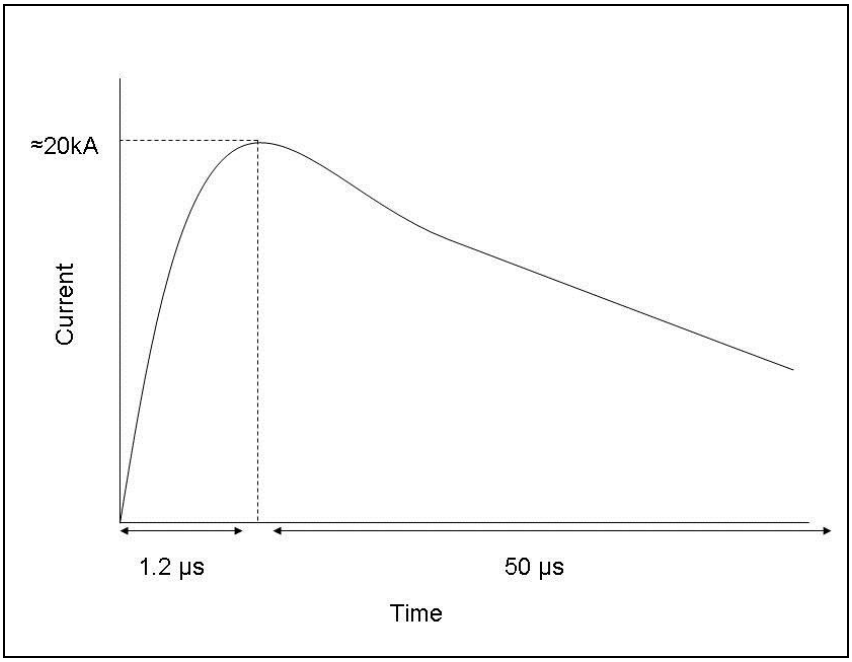


Figure 1.9 Lightning as current source

The current as a function of time (fig 1.9) has been shown to follow the relationship in (1.4) [8].

$$I(t) = I peak(e^{-\alpha t} + e^{-Bt}) \qquad (1.4)$$

This produces a high overvoltage impulse that travels out in both line directions from the point of impact. Such overvoltages can lead to flashover between any live and grounded metallic parts of the line. However, arcing usually occurs over the insulator strings on transmission towers since this is the shortest path to ground. This process is encouraged in the case of high overvoltages by arcing horns. These are two conductors separated by a gap that are placed in parallel with the insulator. They are deliberately designed to take some of the stresses from the insulator strings, and send most of the energy to ground. Unfortunately, this process must necessarily lead to a short circuit fault as the current flows through the arc to ground. The faulted phase must be de-energised by a circuit breaker in order for the arc to be extinguished. Once arcing has extinguished, and the products from the arc path have cleared, the circuit can be re-energised and the line can continue normal operation – the fault duration is only temporary.

Generally, any event that causes the conductor to break, come into contact with ground or other conductor(s) in a short circuit will cause a permanent fault or semi-permenant fault. Anything that triggers an arc between a conductor and ground, or two conductors will cause a transient fault. On the other hand, permanent faults, for example caused by a tree or a downed line, have a constant resistance. (Although for permanent faults, a much shorter arc will occur on breaker opening). This key difference between fault types makes adaptive autoreclosure a viable concept.

1-5 Autoreclosing

Autoreclosing is the practice of re-energising a circuit shortly after a fault has occurred. It takes advantage of the fact that on overhead lines, 80-90% of faults are transient in nature [4]. An autoreclosing relay will send a signal to a circuit breaker to reclose, re-energising the faulted circuit. Should the fault persist after this time, the protection will operate again and retrip the line, and usually the autoreclose will be disabled and the fault assumed to be permanent. In some schemes, particularly at distribution level, the autoreclose relay will be calibrated to attempt to reclose a number of times, known as multishot reclosure. There are a number of parameters involved in configuring autoreclosing relays, of which the most important are dead time, reclaim time and number of shots. Dead time is the time after the protection operates before the reclosing relay attempts the first reclosure, and the reclaim time is the time from the first reclose signal until the relay is ready to respond to further faults. The number of shots is how many reclose attempts are allowed before lock out, usually set at increasing time intervals between shots.

The application of autoreclosing depends on a sequence of events involving the operation of the circuit breakers, the protection relay and the autoreclose relay. The way in which these elements interact with each other can be visualised on the timelines in figure 1.10 and 1.11. Figure 1.10 shows how these system elements respond to a transient fault and figure 1.11 a permanent fault. The scheme depicted here is for a single shot autoreclosing relay, in that the autoreclose is disabled after the first failed attempt.

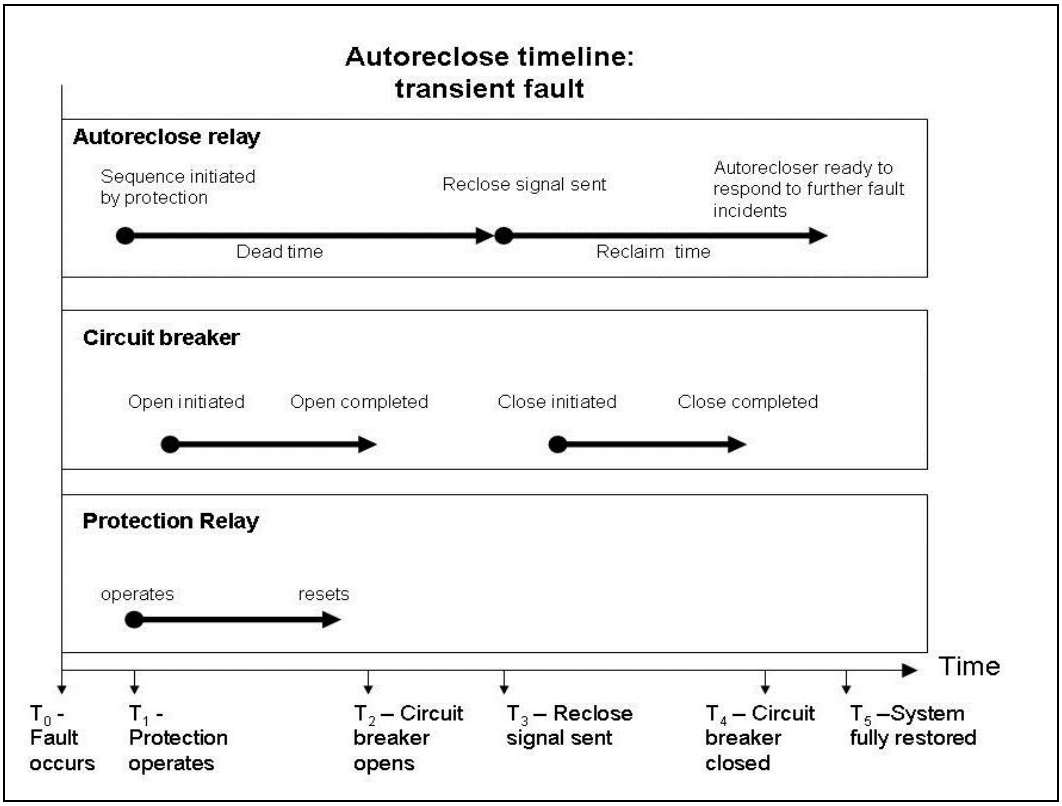


Figure 1.10 Autoreclose timeline for transient fault

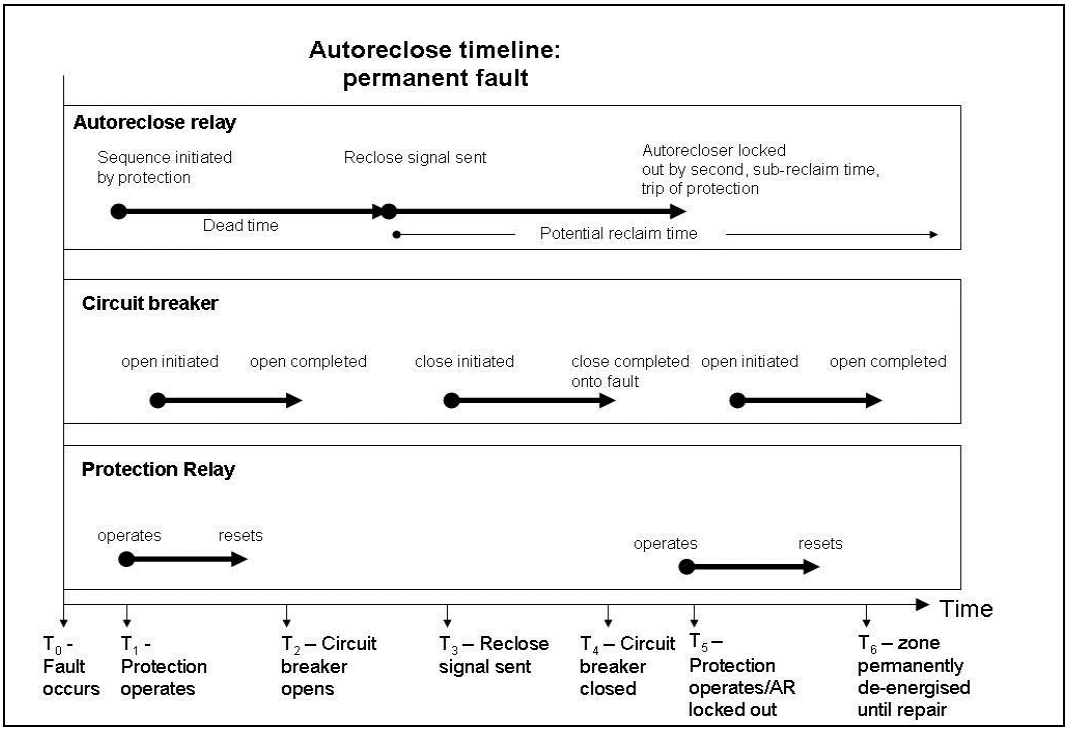


Figure 1.11 Autoreclose timeline for permanent fault

Settings for autoreclosing relays are heavily dependent on the system topology, switchgear, local protection relays and voltage level. At distribution level, one must also consider how different consumer's loads may be affected by autoreclosing.

Reclosing clearly brings advantages in terms of continuity of supply. In loosely connected transmission systems, system stability can also be improved. The drawback is unnecessary secondary shocks to the system caused by failed reclose attempts. This can happen when a reclosure is made onto a permanent fault, or a transient arcing fault is not given sufficient time for the arc to extinguish and for the air in the arc path to deionise. The approximate relationship between minimum arc deionisation time and system voltage shown in figure 1.12.

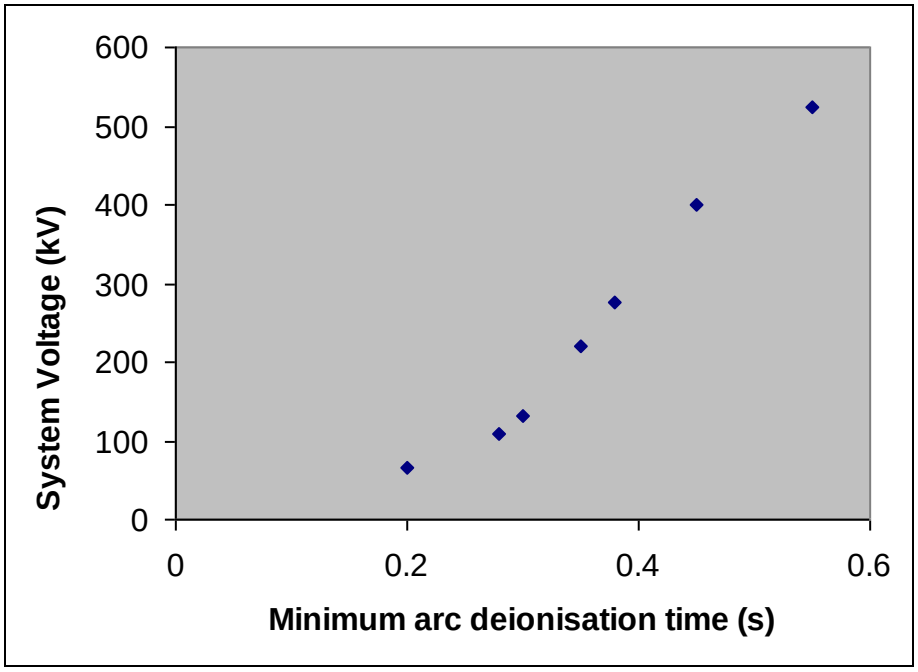


Figure 1.12 Minimum arc deionisation time

The redistribution of energy caused by reclosing attempts at transmission level can cause further damage to nearby equipment. In some cases, it is reasonable to consider no reclosing preferable to the chance of a failed reclose attempt.

For example, particular care must be taken for generators. The potential mismatching of mechanical and electrical power can lead to torsional stress and consequent damage to drive shafts. This occurs when a step change in the electrical network causes the

generator to accelerate or decelerate rapidly the sink of electrical energy because the sink of electrical energy is suddenly removed or introduced. Care is taken over coordination of protection for generators, and is such that mechanical stress is minimised for grid faults. However, when autoreclosing is applied on the electrical network, careful consideration must be taken as to how generators may be affected, and how the loss of generators may affect grid frequency. Mechanical considerations apply for large industrial machines acting as motors.

Single pole tripping in transmission

In the UK, a policy of delayed autoreclosure (DAR) is currently used. In this scheme, all three phases are tripped after a fault. Following a dead time of a number of seconds and a synchronism check, autoreclose is attempted. The tightly connected system is unlikely to fall out of synchronism in this time, and the delay allows ample time for the arc to deionise and the resulting power swings to decay, leading to better stability following the reclosure.

In weakly connected systems, single pole tripping coupled with fast autoreclosure may be desirable. This is because the separate sections are likely to lose synchronism more quickly, as there is no route for the restorative forces between synchronous generators to travel. For an increased cost in switchgear, and additional phase selection element in the relay, it is possible to only open the faulted phase, leaving the other healthy phases in service. The healthy phases continue to transmit synchronising power across the circuit until the transient fault has cleared and the faulted phase may be brought back into service by fast reclosure. This takes advantage of the much higher incidence of transient, single phase to ground faults over any other type of fault, but cannot be used for permanent faults or other fault configurations. If other faults are detected by the protection, usually the logic is set to trip all three phases, and attempt a three phase reclose after an increased interval.

In single pole tripping the mutual coupling between the healthy phases and faulted phase sustains a lower current fault arc on the faulted phase. This is known as the secondary arc. It is unsafe to close whilst this arc persists or the fault will re-establish itself. The ionised arc path must also be given time to clear or the arc will restrike when the circuit is re-energised. The speed at which the deionisation occurs is

dependent on a complex interplay of parameters, such as fault current, line length, wind speed, system voltage, fault type and coupling from nearby conductors. Secondary arcs considerably extend the minimum deionisation times shown in figure 1.12.

In general, autoreclose elements add a greater degree of complexity to the protection and great consideration must be placed as to whether the advantages outweigh the drawbacks for all scenarios that may occur. Should autoreclosing be favourable, the settings and coordination of reclosing is heavily dependant on the local system topology. For example, the order in which circuits and connecting buses are reconnected depend on local generation. Careful consideration must be placed on how the circuit is reenergised so that there is no large discrepancy in any quantity when the circuit breaker closes.

Adaptive autoreclosure

A very attractive possibility is making a real time diagnosis as to whether the fault is permanent or transient. In the case of a permanent fault, the autoreclose may be disabled. Furthermore, in a truly online scheme, the duration of transient faults may be determined and autoreclosing only permitted after the fault has been cleared. This technique is known as adaptive autoreclosure. For brevity, the general concept of adaptive autoreclosing may be referred to as AA hereafter.

Adaptive autoreclosure prevents unnecessary, secondary shocks to the system by avoiding failed reclose attempts. The most desirable corrective action is taken under the circumstances. When a system is stressed, due to weather conditions, unprecedented demand, etc, this helps to minimise the chance of cascading line trips due to overloading of adjacent power corridors. AA has positive consequences stability of the system, as well as continuity of service and security of supply. These are examined in detail in the next chapter.

Chapter 2 – Literature Review on Adaptive Autoreclosure

2-1 Introduction

This chapter looks at the concept of adaptive autoreclosing in detail through the existing literature. The history of autoreclosing, different approaches to determining reclosure deadtimes and diagnosis between transient and permanent faults are discussed. The chapter concludes by highlighting the literature gap involving electromagnetic transient studies and wind turbine technology.

2-2 History

Autoreclosure came into wide spread use on power networks in the 1920s, but many companies began manufacturing specialist autoreclosing electromechanical devices in the preceding decade. For example, automatic switching operations were brought in on US electric railways as early as 1913 [12].

In the 1930s, improvements in circuit breaker design made high speed autoreclosure a possibility. Multi-shot autoreclosure became general practice over the next twenty years. This would usually involve three reclose operations – the first one at high speed and then a further two delayed shots. A 1954 study reported 90% of first reclosure attempts to be successful, 4% of second attempts and 1% third attempts. The remaining 5% of faults failed to clear [13].

Although the technology improved over the years, the basic autoreclosure philosophy remained the same. Microprocessor based autoreclosing relays of the 1980s and 1990s were far smaller, incorporating all the necessary equipment into a single device. In recent years the need for stand alone autoreclosure has diminished as they have become built into multifunctional numeric relays.

2-3 Autoreclosure considerations

A real world autoreclosure scheme for transmission lines must take into account many factors [12]. These all assess whether it is advantageous to allow autoreclosure given the local network, and if so, what would be the optimal settings in the event of a fault.

Important considerations in planning for autoreclosure include, but are not limited to:

a) Locality of generators (particularly in three phase autoreclosure), since unsuccessful reclosure attempts can lead to damage through torsional forces
b) The criticality of loads connected to the line
c) Coordination with other protection relays, for example, faults on other equipment within the protected zone (most faults on transformers, underground cables and switchgear are permanent and should have reclosure blocked)
d) Capability of the circuit breakers – single pole facility, reaction time etc
e) In distribution systems, coordination of reclosing when the line has other branches

In addition, a number of time varying factors may dictate whether autoreclosure should be blocked for particular operating conditions.

a) Recent fault history – this could indicate the presence of a recurring transient fault, for example, due to a tree limb being repeatedly blown onto the line
b) Maintenance being carried out on energised lines – in this case, autoreclosure should be blocked for safety reasons
c) Weather conditions i.e. settings may change during a thunderstorm

High-speed autoreclosing is generally that which allows attempts to reclose in the quickest possible time. This is usually pre-set at around 20 cycles or 400ms, but varies depending on voltage levels and can be estimated with the formula (2.1):

$$t = 10.5 + (V_{l-l})/34.5 \qquad (2.1)$$

Where t is time expressed in cycles. (Formula (2.1) is taken from IEEE [12], with the power frequency is at 60Hz).

In high speed autoreclosing, a synchronism check is not normally necessary, meaning that the voltage phase and magnitude difference at buses at either end of the line are kept within acceptable limits. In the case where the first reclosure attempt is successful, this has desirable implications for stability and continuity of supply.

There are other approaches to conventional reclosure where it is advantageous to have a delay between the fault and the first reclose attempt. The conditions for this delayed autoreclosure (DAR) can also be adaptive and governed by a number of factors. But for the purposes of this thesis, hereon in "adaptive autoreclosure" refers to both:

- reclosure in the quickest possible time for a transient fault
- or blocking the reclosure altogether for a permanent fault

2-4 Early work on adaptive autoreclosing

The concept of adaptive autoreclosing was proposed by Rockefeller, *et al* in a 1988 paper *Adaptive transmission relaying concepts for improved performance* [14]. This paper proposes adapting the dead time of an autoreclosing relay to suit system conditions. A method of detecting sympathy trips is presented, and the authors suggest high speed autoreclosing should only be enabled following detection of a sympathy trip or during thunderstorms where the chance of lightning strikes is high.

The 1990 paper *Adaptive Automatic Reclosing* [15], describes a method of diagnosing a transient or permanent fault by closing one of the phases and measuring the resulting voltages. These voltages are compared to simulated cases to determine whether a fault persists, and then the remaining phases are allowed to reclose or reclosing is blocked if the fault is permanent. This method requires a diagnostic reclose operation which in some cases may make the situation worse. For example, the reenergised phase may prove to be the faulted phase, or the reclosure may cause the fault to evolve to include this phase. Although this problem is mitigated by closing the phase that is least likely to involve the fault, it is possible to envisage scenarios

where this methodology is less beneficial to the system than conventional autoreclosure.

2-5 Diagnosis of transient and permanent faults.

As mentioned in Chapter 1, generally speaking, transient transmission line faults involve arcing and permanent faults do not. Arcing faults have a dynamic resistance whereas permanent faults have a fixed resistance.

Diagnosis between transient and permanent faults using signal processing alone was first proposed in the early nineties when it became possible to model the arcing signature associated with a transient fault using EMTP software, and its derivatives.

The dynamic resistance results in a high frequency signature superimposed onto the fundamental voltage sinusoid, whereas in the permanent fault case, this is not present. Figures 2.1 and 2.2 illustrate the difference between a permanent fault and an arcing fault respectively. They show faulted phase voltage seen by the relay at one end of the transmission line, on the line side of the circuit breaker.

These fault cases were generated by ATP draw. (ATP draw, Alternative Transients Program, is the graphical freeware version of the EMTP software). The system in question was a 128 km transmission line with 5GVA and 35GVA capacities at sending and receiving ends respectively, modelled as voltage sources behind an equivalent sub-transient reactance. The various modelling techniques are discussed in chapter 4 of this thesis and these figures are included at this stage for illustrative purposes only.

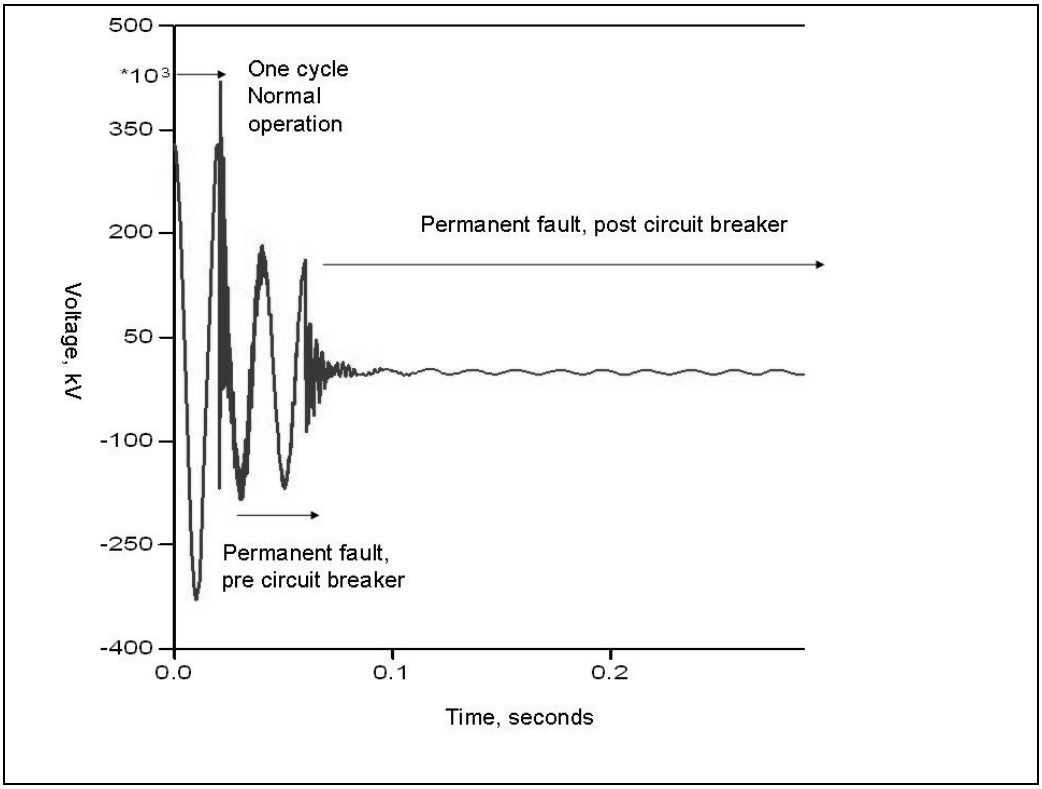


Figure 2.1: Variance of sending end voltage with time fixed resistance, seconds and volts x 10^3

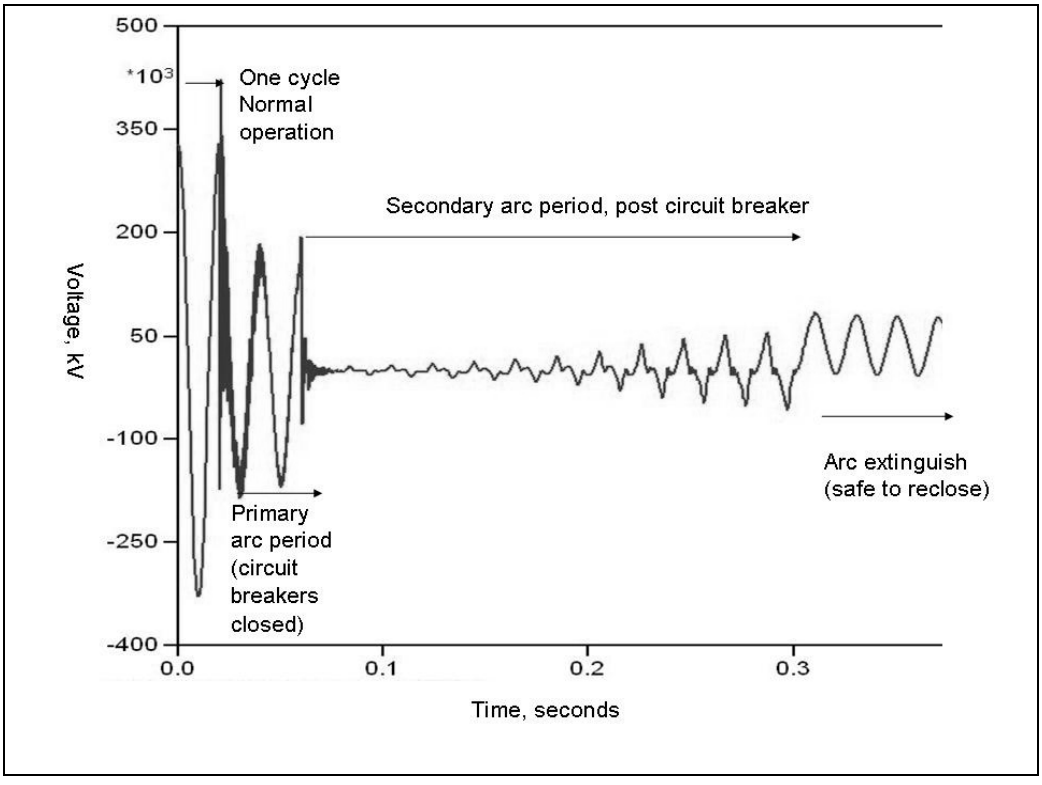


Figure 2.2: Variance of sending end voltage with time realistic arc model, seconds and volts x 10^3

2-6 Adaptive autoreclosure for single pole circuit breakers

Even though a fault may involve any combination of phases and ground, by far the most common fault is a single phase to ground - about 70-90%. Adaptive single pole autoreclosure (AdSPAR) is the technique when a single phase is tripped following a single phase to ground fault. These methods rely on single pole tripping since they must check for the existence of a secondary arc, and then if a transient fault is diagnosed, they must detect the extinguishing of the secondary arc to determine when it is safe to reclose. Clearly, this technique requires a circuit breaker capable of tripping only a single phase. On single circuit lines deploying three phase tripping, the secondary arc does not exist since there are no energised phases to drive it through mutual capacitive and inductive coupling.

The detection of the secondary arc is fairly robust because the faulted phase voltage waveforms are very different in the case of a permanent fault from that of a transient fault. Researchers have therefore presented a number of methods to diagnose the existence of a secondary arc. These methods use frequency transforms, AI or numerical techniques or a hybrid of these.

The most common approach is using some form of windowed frequency transform on the incoming voltage waveform for feature selection, and coupling this to an AI method such as artificial neural networks or fuzzy logic. Fitton et al present a much referenced method in [16] and [17] where a windowed discrete fourier transform decomposes the time signal into different frequency boundaries, and this is fed into a artificial neural network. The basis of this technique is shown in figure 2.3.

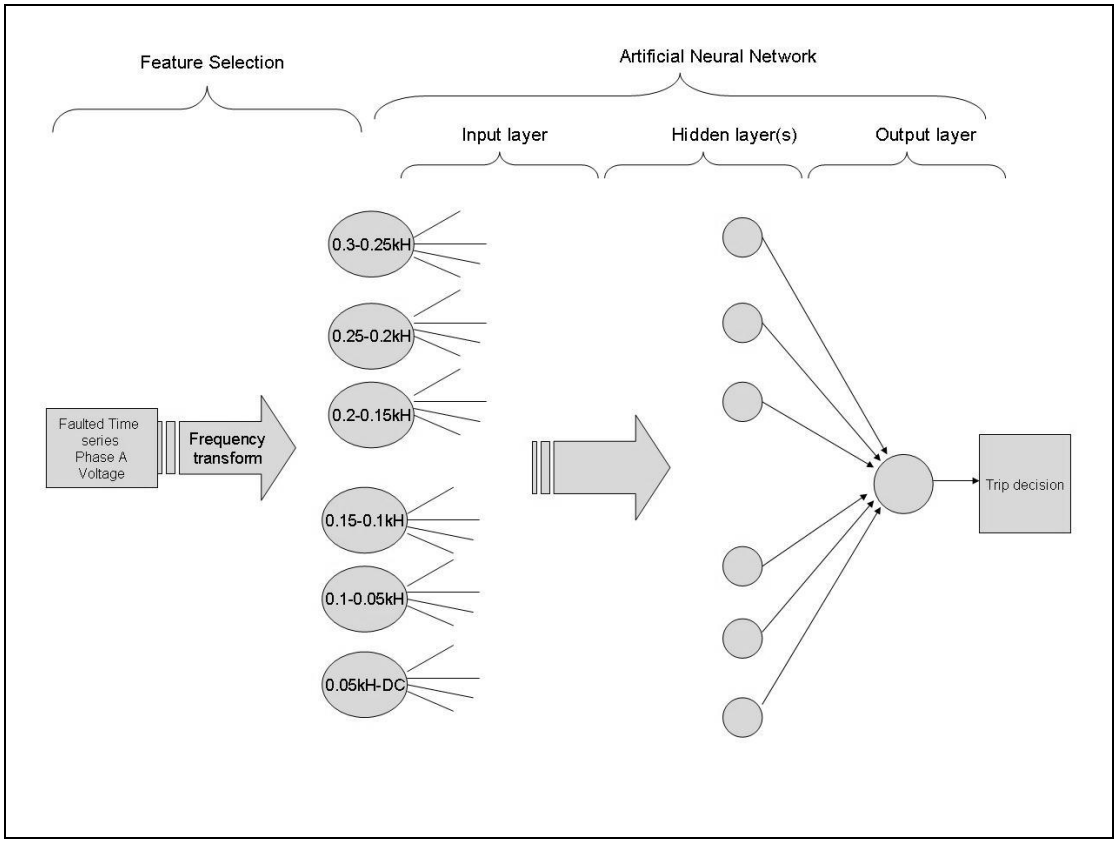


Figure 2.3: Fitton et al method of AdSPAR

Most AdSPAR schemes rely on high frequency information for fault diagnosis. As such, the use of digital signal processing is mandatory in obtaining this information from the frequency domain. The traditional Fourier transform is not suitable for localised transient signals since all information from the time domain is lost. The short time fourier transform (STFT) is an improvement since it retains some time information and some frequency information. Events can be localised and approximated in the time and frequency domains to varying extents. However, the STFT always requires a compromise between the resolutions of these two quantities – time resolution must be sacrificed in return for frequency resolution and vice versa. The discrete wavelet transform (DWT) has the advantage of varying time and pseudo-frequency resolutions. This property makes it ideal for power system transients, where it is common to have high frequency events, specific in time, imposed on a continuous low frequency power signal. The effectiveness of the wavelet transform is related to the extent the mother wavelet resembles the original signal. For this reason the Debauchies 4 wavelet is often used to diagnose arcing behaviour. AdSPAR discrete wavelet transform schemes are presented in [18] [19] and [20].

However, direct comparison of the DWT and the STFT has shown that the short time fourier transform is more robust when used for feature extraction when applied to the Fitton et al artificial neural network technique [21] A more extensive explanation of neural networks can be found in chapter 5. Briefly, the network is a fully connected feed-forward network known as a multilayer perceptron. Time-varying magnitudes of the frequency bands form the input layers of the perception. The network has one or more hidden layers, and these map to a single output neuron which gives a decision as to whether the fault is transient or permanent.

Alternative numerical algorithms based on measurements taken at the power frequency are presented in [22] and [23]. These papers present a technique using least mean square error estimation to determine the presence of an arc voltage. Although this method is computationally less expensive than methods involving spectral analysis, it relies on the assumption that a permanent fault has no resistance. Also it is only tested for three phase faults, which are the least common type of faults and when they do occur, and are very rarely transient. Reference [24] presents an AdSPAR scheme based on the percentage total harmonic distortion (THD), which is far greater whilst the secondary arc exists. However, this method also relies on the frequency transform to determine the harmonics up to the 5th order.

Of all the techniques for AdSPAR, the method developed by Fitton et al [16] is the only one that has been deployed and proven on a real world system, and documented in the literature. In [25] the ANN technique is shown to be successful on a 275kV line on the South African supergrid. This network is relatively loosely connected so makes use of single pole autoreclosing. Utilities are often reticent about deploying ANN protection methods on safety critical systems, because they are considered to behave in a non-deterministic way if the input data varies significantly to the training data. However, it should be considered that in the Fitton technique, the ANN is used to derive discrete binary value rather than a value on a continuous range. In other words, if the time averaged output of the network is below a certain value then it is safe to reclose [25]. Such concerns are also addressed, in this application, by testing the trained ANN response for all possible inputs. Furthermore, in this paper, the consequence of a wrong decision are discussed and assessed to be no worse than conventional autoreclosure.

2-7 Three phase adaptive autoreclosing

As mentioned previously, all AdSPAR techniques rely on the ability of the circuit breaker to trip a single phase, and are clearly only applicable to single phase to ground faults. However, in the UK, the current practice across transmission networks is to use three phase tripping in conjunction with delayed autoreclosure (DAR), whereby the first reclose attempt is made at around 20 seconds. This approach has been adequate for many years since the UK grid is quite strong and secure, and the overriding consideration is the prevention of unsuccessful reclose attempts.

In systems deploying three-phase tripping, it is only possible to use the primary arcing period to diagnose the nature of the fault, since the voltage and current signals attenuate very quickly on all phases post circuit breaker operation. This is difficult since the primary arc period looks very similar to a permanent fault on the faulted phase. This is the case because over this period, the nominal time averaged arc resistance is very similar to the permanent fault resistance (both in the region of 0 – 100 Ω). In [26] the authors develop an adpative three phase autoreclose (AdTAR) technique for double circuit lines. In the case of double circuit towers, predominant in the UK, inter-circuit coupling is enough to drive a secondary arc in the same way that inter-phase coupling does in single pole circuit breaking. Given this is true for the line in question, the AdSPAR ANN can be retrained to recognise arc extinction in the three phase tripping case. This method of course relies on the healthy circuit being initially energised and remaining so following the fault.

An investigation by Websper et al [27] has shown that reclosing immediately after secondary arc extinction does not necessarily guarantee a successful autoreclosing event. This is due to dither region where the conditions may cause the secondary arc to restrike, if the voltage rises above a certain level. This is in the region of two power system cycles (at 50 Hz) which is the response time for most modern circuit breakers.

2-8 Optimal autoreclosure

Given the incidence of a transient fault and the desirable decision to reclose, there is then further consideration as to the optimal moment to reclose. In [28] an algorithm is

presented to calculate the optimal reclosure time. Here the primary system modelled consists of a synchronous generator feeding an infinite bus bar through a connecting overhead line. The authors state that the optimal reclosure point is when the power angle is closest to the pre-fault point. In this case, the reclosure adds extra damping to power angle oscillations and they attenuate very quickly to a stable steady state.

Along with system stability, it is important to consider the effect of autoreclosing on local generators [29]. Large stress oscillations arise due to the mismatching of mechanical power with electrical power following the initial fault disturbance. These may be exacerbated by fast autoreclosing, particularly if the autoreclosure imparts a torsional impulse that is in phase with the existing oscillations [30]. Another factor to optimal autoreclosing is to minimise fatigue on generator drive shafts.

Recently, a multiagent systems (MAS) based approach was developed to combine optimal and adaptive autoreclosing [31]. In this method, one agent determines the arc extinction, another determines the optimal reclosure time and a third assesses the transient energy of the system following a circuit breaker event. The agents work together to apply sequential autoreclosing at the moments when conditions are most favourable.

In practice, there are many issues that complicate adaptive autoreclosing. The system's response across the entire frequency spectrum will be affected by many system parameters other than just arcing. These include the transmission line conductor configuration, electromechanical oscillations from nearby generators, harmonics injected by power electronics, FACTs devices [32], and interference from communication systems. It is of utmost importance therefore that the primary system is modelled to a high accuracy to verify that these factors can be accounted for on the system on which these methods are to be deployed.

2-9 Adaptive Autoreclosing and Power Electronics

One particular issue that remains unanswered is the effect of renewable generation technology on local transmission line transients, and thus adaptive autoreclosing techniques. In recent years, environmental concerns over burning fossil fuels have led

to a rapid growth in wind generation world wide. In the period between 1995 and 2006, installed global capacity grew from 4,800 MW to 59,000 MW representing a 1229% increase over 11 years [3]. At the time of writing, the installed capacity in the UK is 5,111 MW installed in a mixture of onshore and offshore projects. Wind turbine technology has also moved on considerably. Maximum turbine sizes have grown from 600 kW, 60 m diameter in 1995, to the considerable 5MW, 124m diameter by 2004.

The technology in the wind generators has also evolved to deliver this gain in capacity. Turbines have evolved from fixed speed, whose power fluctuations are directly exported to the grid, to variable speed that have complex power electronics to maximise output and control of active and reactive power.

Early issues with power quality, voltage stability and flicker have been largely overcome with power electronic solutions. However, power electronics introduce high harmonics onto the grid, which may interfere with high frequencies associated with autoreclosure schemes [33]. Variable speed turbines are discussed in detail in the following chapter.

In the past, it has been sufficient to completely disconnect wind farms following a fault on the network. However as larger windfarms become connected at higher voltage levels, they must remain connected under fault conditions, in order to maintain the system frequency within acceptable limits. This so-called ‘ride through’ capability, which is part of the grid code, or technical specifications for connection, presents several issues for power system protection. Since variable speed wind turbines use Voltage Source Converters (VSCs), their contribution to fault current is different to equivalent conventional synchronous plant. This is because the mechanical assembly and electrical frequency are partially or completely decoupled by the power electronic converter. This has consequences for overcurrent and distance protection, and it is important that protection engineers account for these when configuring relays. This also has bearing on the design of novel transient based protection, including adaptive autoreclosing.

The growth in wind technology has seen a concomitant increase in the literature. There have been a multitude of studies, for example, [34] – [37] conducted on different aspects of wind farms, and their impact on the grid. These range from modelling of individual turbine technologies to the effect of wind farms on voltage stability and their response to grid faults. However when dealing with transient short circuit fault tests, they rarely deal with unbalanced faults (i.e. single phase to ground faults) or include a realistic transmission line model. Moreover, results are often measured at the point of connection to the grid rather than at the local substations. Clearly, when assessing the impact of wind technology on power system protection, it is important to focus on what the relay 'sees'.

Given the predominance of power electronics in wind turbines, it would be useful to ascertain their effect on novel transient based protection. Virtually no studies exist to investigate the importance of wind turbine technology on this emerging area of protection, particularly adaptive autoreclosure and therefore this presents an important research question and is a major contribution of this thesis.

Chapter 3 – The effect of wind generation on transmission line short circuits

3-1 Introduction

As mentioned in chapter 2, the existing literature does not deal with the effect of wind farms on adaptive autoreclosure. Specifically, the effects of wind farm power electronics are likely to have some significance due to their introduction of harmonics onto the grid. This part of the thesis presents an investigation into the effect of wind farms on signatures used in adaptive autoreclosure. Using DIgSILENT's power factory software, a small part of the UK Generic Distribution System is constructed as a test system. Electromagnetic transient simulations are conducted on a number of parameters known to affect fault sequences. These are processed using the Discrete Wavelet Transform in Matlab to compare the signals at different frequency bands. Wind technology is shown to have less significance than key parameters, and the remainder of the thesis aims to prove the results of this investigation.

3-2 Wind turbine types

Wind turbine types

There are three basic wind turbine types in existence today. They can be grouped into fixed speed and variable speed.

Fixed speed turbines are the oldest and simplest designs. Their rotational speed is fixed regardless of wind speed, and determined by the supply grid frequency, gear ratio and generator design. These turbines have optimized efficiency for one wind speed. (Some of these designs include two winding sets in their generators, one for low speeds and a set with fewer poles for high speeds). The major drawback of fixed speed operation is that wind speed fluctuations are exported to the grid as fluctuations in electrical power. Also, reactive power control is not possible, although a fixed capacitor bank is usually included to compensate for consumption of reactive power.
Fixed speed turbines employ three different overspeed breaking methodologies to prevent damage at high wind speeds. These also serve to reduce fluctuations in power,

but do not eliminate them. In ‘passive’ stall, the blades are aerodynamically designed to lose power at high wind speeds. Pitch control is moving the angle of the blades out of the wind at high speeds. Active stall control is a combination of both, in as much as the stall of the turbine is actively controlled by pitching the blades far into winds at high speeds.

As the permeation of wind generation has increased, the rules of connection to the grid have grown stricter to mitigate adverse consequences due to power fluctuations. The variable speed turbine has evolved to cope with these regulations. In recent years, two main types have emerged to dominate the market, both of which are pitch control. As of 2002, the DFIG constitutes 46.8% of the market share and the Full Converter (FC) 20.3%, although in subsequent years, the FC type is beginning to increase due improvements in power electronics [33].

DFIG

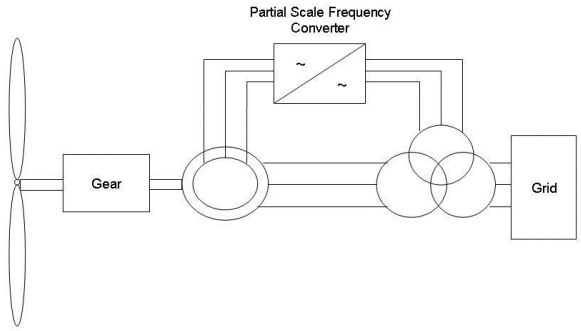


Figure 3.1: Schematic of DFIG Design reproduced from [33]

The term DFIG stands for “doubly-fed induction generator” and its basic design is shown in figure 3.1. It consists of an induction generator, whose rotor is connected to a partial frequency converter via slip rings, which in turn is coupled to the grid through a three-winding transformer. The stator of the generator is connected directly to the grid. The power electronic converter makes up for the shortfall or excess speed difference (and thus difference in turbine’s mechanical frequency and grid’s electrical frequency) by injecting the appropriate variable current into the rotor. In the over-synchronous case, power flows from the rotor to the converter to the grid, and in the

sub-synchronous case, it flows in the opposite direction. In either case, however, net power flow is onto the grid via the stator. This mechanism enables the turbine to operate at a wide range of speeds, typically up to +30% and -40% of synchronous frequency.

The power electronics in the frequency converter enable control of active and reactive power, and can even be utilized by the grid to actively assist in frequency and voltage control. Greater operational range of speeds can be achieved at the cost of heavier duty, and therefore more expensive, power electronics. A drawback of the DFIG is that it requires slip rings to connect the rotor to the rectifier, resulting in losses and increased maintenance.

Full Converter

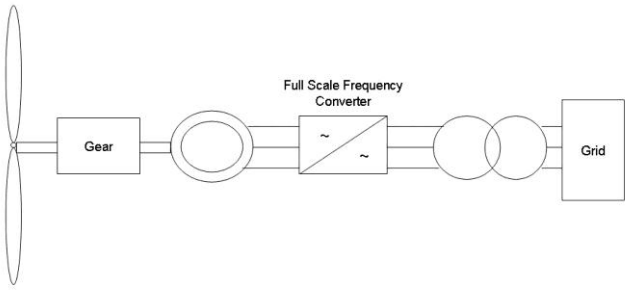


Figure 3.2: Schematic of Full Converter Design reproduced from [33]

The most modern design is only economically viable due to advances in power electronics. In the case of the full converter (FC), shown in figure 3.2, a fully rated frequency converter is used. The frequency converter handles the full power of the turbine and maintains synchronism with the grid frequency over a full range of turbine speeds. It also acts as a buffer against grid side transients to the generator and power fluctuations onto the grid due to gusting conditions. The grid is de-synchronized from the generator so either an induction generator (usually squirrel cage) or a synchronous generator (usually wound rotor) can be used. The induction generator is less expensive but requires a magnetizing current for start up, which must be supplied from the grid.

Power electronics

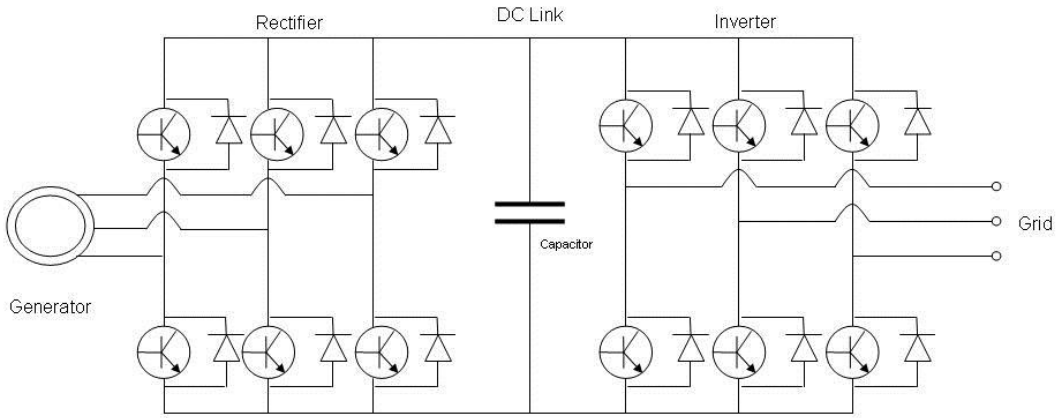


Figure 3.3: Back to back frequency converter [33]

All wind turbine types use power electronic devices such as soft starters, capacitor banks and frequency converters. Of these, most integral to operation is the frequency converter, which is included in both variable speed designs. The most common topology is the back–to-back converter, shown in figure 3.3. This allows two-way power flow, and consists of a rectifier and an inverter whose transistors are Pulse Width Modulated. Fast discrete switching of these transistors, controlled by modulating the width of signal pulse approximates a DC input signal to sinusoidal AC and vice versa.

The basic premise is that power is taken at one AC frequency, converted to DC and then converted back to AC at the required grid frequency. The capacitor across the DC link gives energy storage allowing the control of the rectifier and inverter to be decoupled. This means more complex control of the transistors can be introduced to suit the requirements both at the grid side (i.e. reduce power fluctuations and control voltage) and at the generator side (appropriate excitation currents can be established along with the desired rotor speed) [39].

In this context, the term ‘transistor’ encompasses a wide range of devices, the particular transistor used in most modern turbines is the insulated gate bipolar transistor (IGBT), a modern variation of the metal oxide semiconductor field effect transistor (MOSFET). The MOSFET is a power semi-conductor device capable of

fast switching. In the “on” state the current flows between the emitter and the collector with very little resistive losses. In the “off” state there is very little leakage current. The IGBT is similar in structure to the MOSFET, but is has an additional PN junction. Clearly, for their significance in power systems, their control and resultant electrical behaviour is more important than their underlying design. Unlike thyristors, power transistors are self-commutated, meaning their switching is independent of the current flowing through them. The main advantage over line-commutated devices is that they are capable of faster switching. The resultant sine wave therefore has a higher power quality, with less lower order harmonics. The IGBT, for example, is capable of switching at rates of 2-20 kHz. The harmonics due to this switching lie at these frequencies or above due the PWM process [40]. Autoreclosing schemes that diagnose the extinguish of the secondary typically only use frequencies up to 500 Hz and are thus unlikely to interfere with the arcing signature. However, it is important to confirm this through detailed EM transient simulation studies, since the holistic impact of wind generation technology may have unforeseen effects on the complex and highly non-linear system.

3-3 Discrete wavelet transform

The adaptive autoreclosing schemes that detect secondary arc extinction normally consist of an initial signal processing stage, followed by a pattern selection algorithm. For signal processing, past schemes have utilized the short time fourier transform and the discrete wavelet transform. Although it is beyond the scope of this chapter to give a full explanation of wavelet transforms, (this can be found in the two-part tutorial [40] and [41]), a brief explanation follows. The discrete wavelet transform offers the advantage of variable time and frequency resolutions. When applied to power system transients, high frequencies are made to favour time resolution. This means high frequency events are specifically localized in time, but have poor frequency resolution. Conversely, low frequency signals have better frequency than time resolution. This property is useful for non-stationary power system transients that consist of localized high frequency information superimposed on the 50 or 60Hz fundamental power signal. It should be noted that the output of the Discrete Wavelet transform frequency is strictly speaking “scale” and time, rather than true frequency; scale can be regarded as a pseudo-frequency.

The discrete wavelet transform takes the form of equation (3.1).

$$DWT(m,k) = \frac{1}{\sqrt{a_0{}^m}} \sum_n x(n) g\left(\frac{k - nb_0 a_0{}^m}{a_0{}^m} \right) \quad (3.1)$$

Where the function g( ) is the mother wavelet and $a_0{}^m$ and $nb_0a_0{}^m$ are scaling terms. The integer parameter, *m*, produces a family of daughter wavelets, and is also responsible for the logarithmic frequency scale. This comes from the term $1/a_0{}^m$, that gives geometric scaling i.e. 1, $1/a_0$, $1/a_0{}^2$, and so on.

The DWT is implemented using different versions of the mother wavelet as high pass and low pass filters. This yields a high frequency signal, the detail cD, and a low frequency signal known as the approximation, cA. Both approximation and details are down-sampled by a factor of two to account for each constituent only containing half the information of the original signal. The detail represents the information at the first and highest frequency band. This first band is defined by the sampling frequency of the original signal, Fs, the upper most boundary being Fs/2 and the lower boundary Fs/4. The upper boundary is a consequence of the Nyquist criterion, which dictates that the highest frequency a sampled signal could properly represent would be half the original sampling frequency. The first approximation itself is transformed to produce the next detail and approximation. This second approximation is in turn transformed again for the next frequency band Fs/4 to Fs/8 and so on. Successive frequency bands get logarithmically smaller, each half the size of its predecessor as the different bands are siphoned off into the details. This process is illustrated for the first three details in figure 3.4. Although the scale or "pseudo frequency" resolution improves with decreasing frequency, the time resolution deteriorates because each new signal contains half the amount of samples as before.

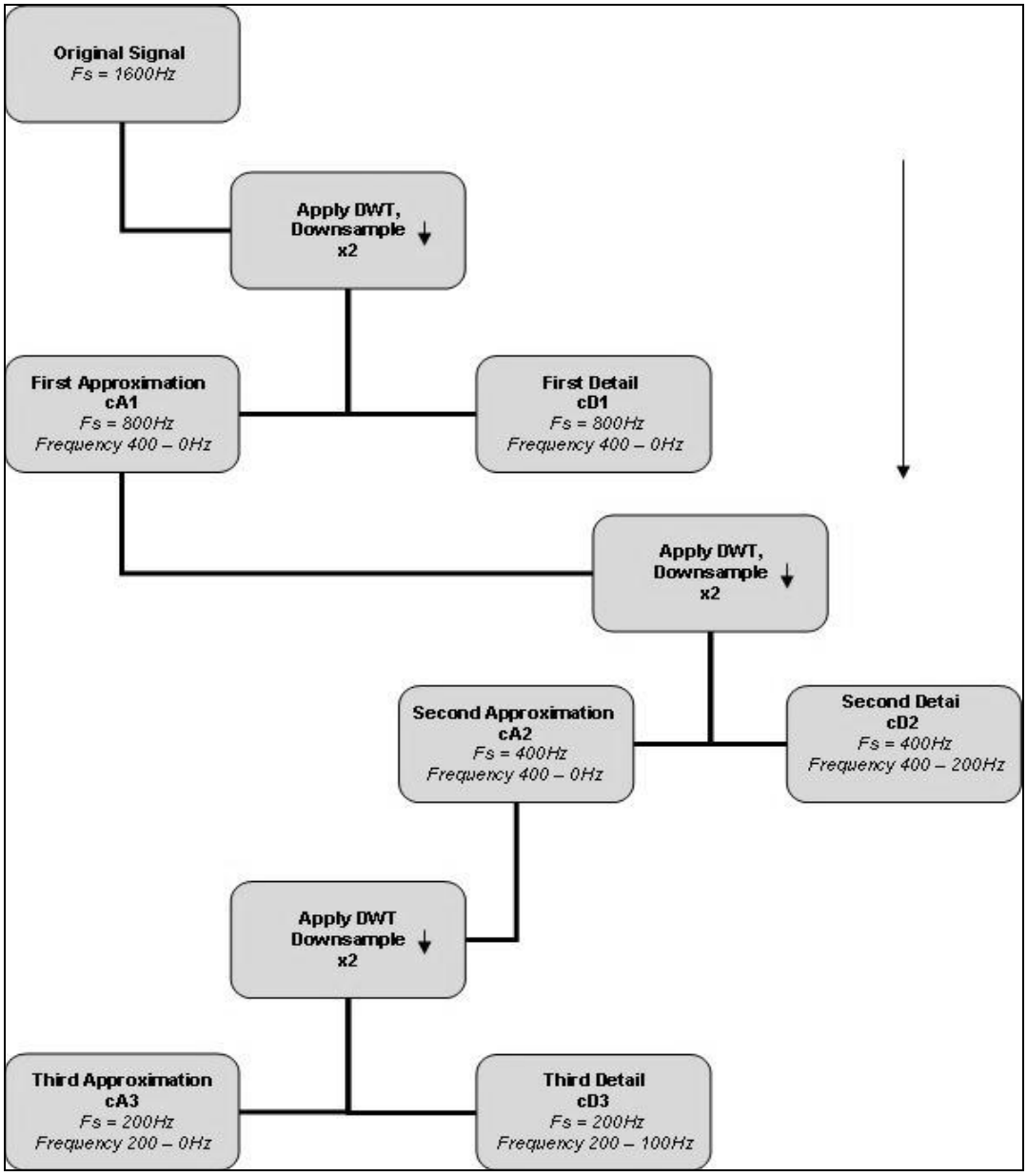


Figure 3.4: Discrete Wavelet Transform algorithm sampling, (frequency values are those used in study)

The best choice of mother wavelet is dictated by the nature of the original signal. Generally a wavelet that most closely resembles the original signal will lead to the best results. The Debaucheries 4 wavelet has been consistently shown to best suit power system transients [42], and so was chosen for this study.

3-4 Study Method

The DIgSILENT software PowerFactory 13.2.336 was chosen for its versatility and in-built wind farm models. PowerFactory is able to handle electromagnetic transient (EMT) studies, deal with load flow considerations and longer-term stability studies demonstrating the viability of the power system over all timescales.

The generating technology was a DFIG based wind farm consisting of 9 x 5 MW generators, and an FC wind farm consisting of 30 x 1.5 MW generators. In all cases, the wind farm was represented by one generator model connected in parallel, using the option in PowerFactory 'number of parallel machines'. Simulations were compared to a base case of 30 x 1.5MW 'bare' synchronous generators without the power electronics or control circuitry of the wind generators. Many generators rather than a single large generator were favourable to be confident that any change in fault signatures would be due to the wind farm circuitry rather than the number of generators. Control circuitry governs output of reactive power in the case of the DFIG and FC turbine models. The generation capacities therefore had to be defined in terms of nominal active rather than apparent power.

I - Test network

The test network used was an approximation of the UK generic distribution system (UKGDS) produced by the BERR funded SEDG centre [43]. The SEDG centre has produced a number of typical networks ideal for conducting studies on distributed generation. The particular network used was the EHV2 "Large rural network" shown in figure 3.5. The large rural network was chosen since bus 7 is a convenient place to connect a hypothetical wind farm. The 13.3 km 132kV branch from 106 to 107 is one of the longest lines in the network representing the likely remote location of the wind farm. Hitherto, adaptive autoreclosure methods have only been extensively investigated for EHV transmission lines of 230kV and above. However, wind farms are usually connected at lower voltage levels at either sub transmission or distribution, and thus basing the investigation at 132kV was more appropriate.

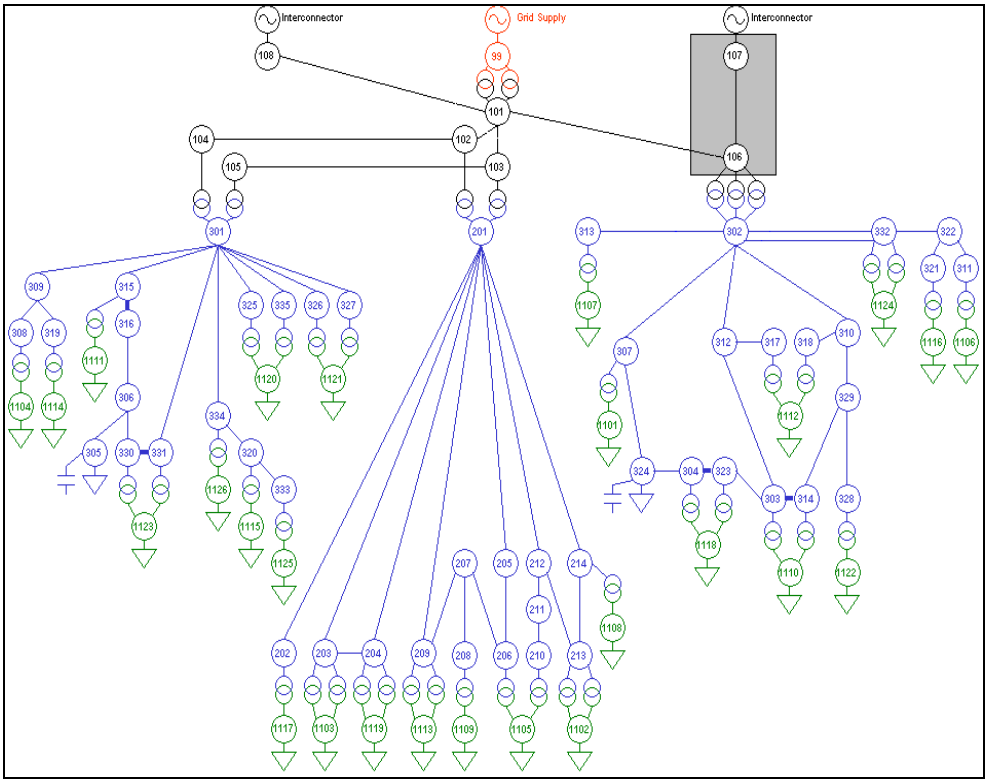


Figure 3.5: UKGDN, EVH-2, Highlighted area used in study

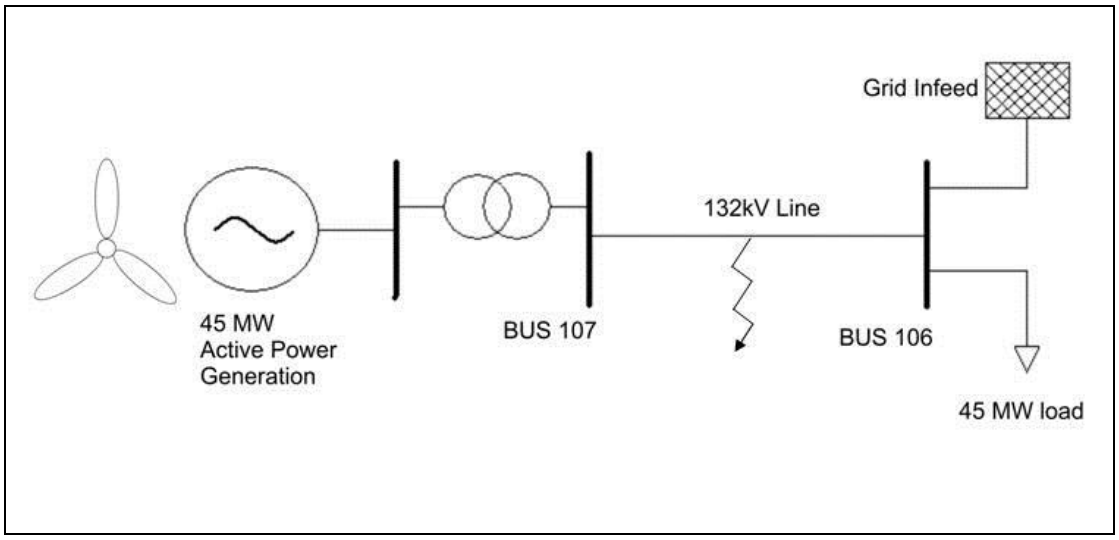


Figure 3.6: Test network, Bus 106 Sending end and Bus 107 Receiving end

With reference to figure 3.6, a load was placed at the receiving end bus. This was to satisfy the load flow simulation component and ensure the line was transmitting power during the simulation. The rest of the grid was represented by Powerfactory's external network component. The short circuit power of the grid infeed was set to a default value of 1 GVA as defined by the UKGDS.

Table 3.1 Line parameters for branch 106 - 107 :

Positive and Negative Sequence Resistance (MVA Base 100)	Positive and Negative Sequence Reactance (MVA Base 100)	Positive and Negative Sequence Susceptance (MVA Base 100)	Zero Sequence Resistance (MVA Base 100)	Zero Sequence Reactance (MVA Base 100)	Zero Sequence Susceptance (MVA Base 100)	Length (km)
0.0475167	**0.1421224**	**0.0312721**	**0.1283494**	**0.4021231**	**0.0186139**	**13.329**

The line model was constructed using PowerFactory's 'Type Line' component, taking the values specified by the UKGDS and shown in table 3.1. The line parameters for the EMT study were calculated by PowerFactory using the in built frequency-dependent distributed parameter model.

II – EMT Simulation

Given the time domain information used to determine autoreclosure may potentially involve both the initial short circuit fault, and the action of the circuit breakers opening, the sequence simulated both these events. The transient simulation was run for one second, the fault inception was at 300 ms and the circuit breakers were set to open at both buses at the nearest current zero after 340 ms, with 2 cycles being a reasonable response time for modern circuit breakers. (In the inception point study these values were altered depending on where the fault was required to occur on the waveform). The default fault type was a single phase to ground fault since this yields the most information by virtue of mutual coupling with the healthy phases. The fault path resistance had a default value of 2 Ω, and was simulated by connecting a phase to ground at a "virtual bus" at the middle of the transmission line. The default site of the fault was at a distance of 6.65 km, i.e. at a point equidistant to both ends.

The electromagnetic transients simulation in PowerFactory was run for 1.1 seconds and at a sampling frequency of 10 kHz. This was a reasonable compromise between speed of simulation and accuracy. The waveforms were subsequently down sampled in Matlab to 1.6 kHz to define appropriate frequency boundaries for the wavelet transform.

3-5 Waveform Indexing

The investigation presented in this chapter generated many waveforms. For brevity, only some waveforms are presented that illustrate the salient points. The following convention was used in data analysis for clarity, and it may be beneficial to inform the reader at this stage. A similar convention is used throughout this thesis.

Each waveform used a five character indexing. The first character indicates current or voltage waveform, I or V respectively. The second character indicates the phase – A, B or C. The third character denotes the bus at which the measurement was taken: sending S or receiving R. The fourth character, C D F indicates the generating technology: conventional, DFIG, Full converter respectively. The fifth character is an alphanumeric value that references the study involved.

For example, IBSD4a would indicate the current waveform, for phase B, from the sending end, using DFIG technology on study 4a while VARF0 would indicate the voltage waveform for phase A, from the receiving end using Full Converter technology, on study 0. Wavelet transform results were also subtitled with their detail level.

3-6 Results

Adaptive autoreclose schemes must be robust enough to cope with the wide-ranging fault conditions that may occur on a line, without affecting the diagnosis in terms of phase selection, fault type or transient arcing time. Such a scheme would require a fail-safe contingency for when diagnosis was in doubt. Some of the most important factors that affect the fault signal are [44]:

I) Generation type
II) Sending end power generation
III) Receiving end short circuit capacity
IV) Length of transmission line
V) Fault type
VI) Location of fault on line

VII) Point of fault inception on wave
VIII) Fault path resistance

The comprehensive approach therefore is to test how each parameter interacts with the generating technology as it varies, and to determine where the differences lie. This establishes the full range of scenarios that the algorithm would be required to deal with.

The default cases, against which all other parameters were varied, were as follows. For all technologies, the base level of generation was 45 MW. Reactive power output is an intrinsic part of the turbine models and so was determined by the simulation. In practice, an autoreclosure scheme would benefit from use of current and voltage information from all phases, and for maximum robustness and reliability could potentially make use of the signals from both ends of the line. Most information however, would be derived from the voltage waveforms on the faulted phase(s), since these exhibit greater disturbance than the healthy phases. The voltage continues to fluctuate after the breakers are opened whereas the current falls to zero. For this reason the study concentrated on the faulted voltage waveform, with additional information supplied by the healthy phases.

I Influence of Generating Technology

The default scenarios are shown in figures 3.1.1 – 3.1.3. The main feature in all default cases is an isolated, high intensity peak across all the frequency boundaries at the fault inception point, and then a smaller, but equally ubiquitous spike at the point the circuit breaker opens. This is due to the near vertical wave-fronts on both fault inception and breaker opening, manifesting themselves in the transform as high intensity features at all frequencies. The high frequency decompositions from the wavelet transforms are shown in figures 3.1.4 – 3.1.6. There are profound differences between the DFIG and the other two technologies. There is considerably more noise in the DFIG case over all frequency bands. This is due to the more complex control circuitry featured in the DFIG model, particularly the crowbar resistance across the rotor windings circuit shortly after the fault inception. The DFIG model automatically implements this at approximately 0.307s, and removes the resistance at 0.807s. The distortion at the wave peaks, following the fault inception, but before the A phase

breakers open (see figure 3.1.3) is not present in the other two cases. This is most pronounced at the 200-100 Hz and the 100-50 Hz intervals, (see figure 3.1.6). The differences between the FC and conventional technologies are unremarkable, with only minor variations in the frequencies after both events, in the 400-200 Hz and 200-100 Hz ranges, see figures 3.1.4 and 3.1.5.

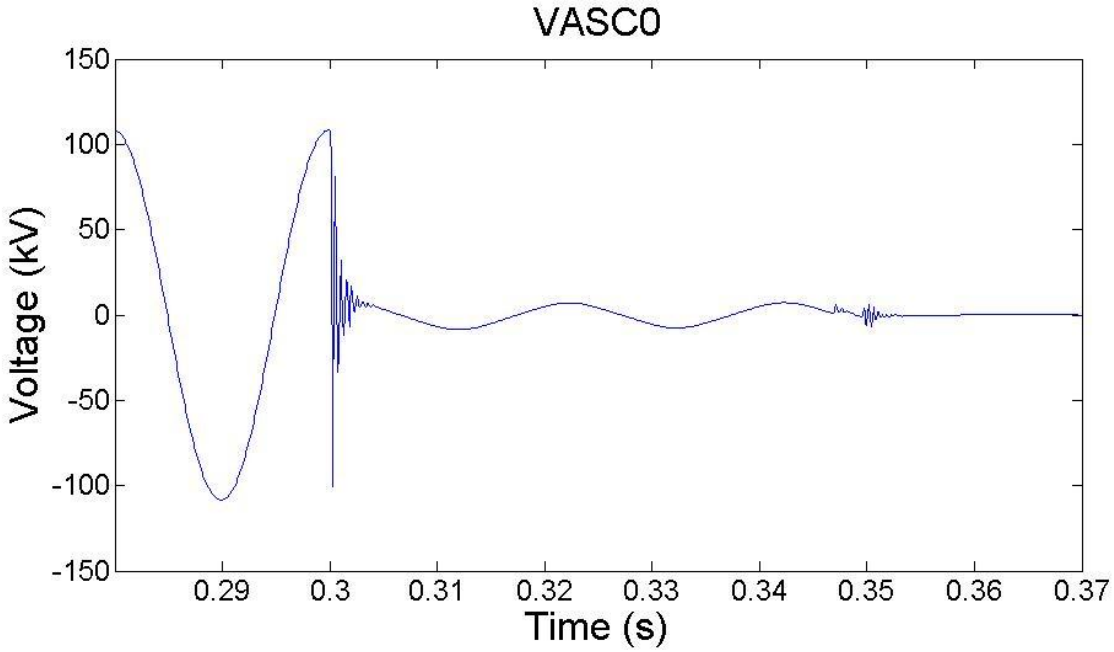


Figure 3.1.1: Faulted waveform for conventional default case

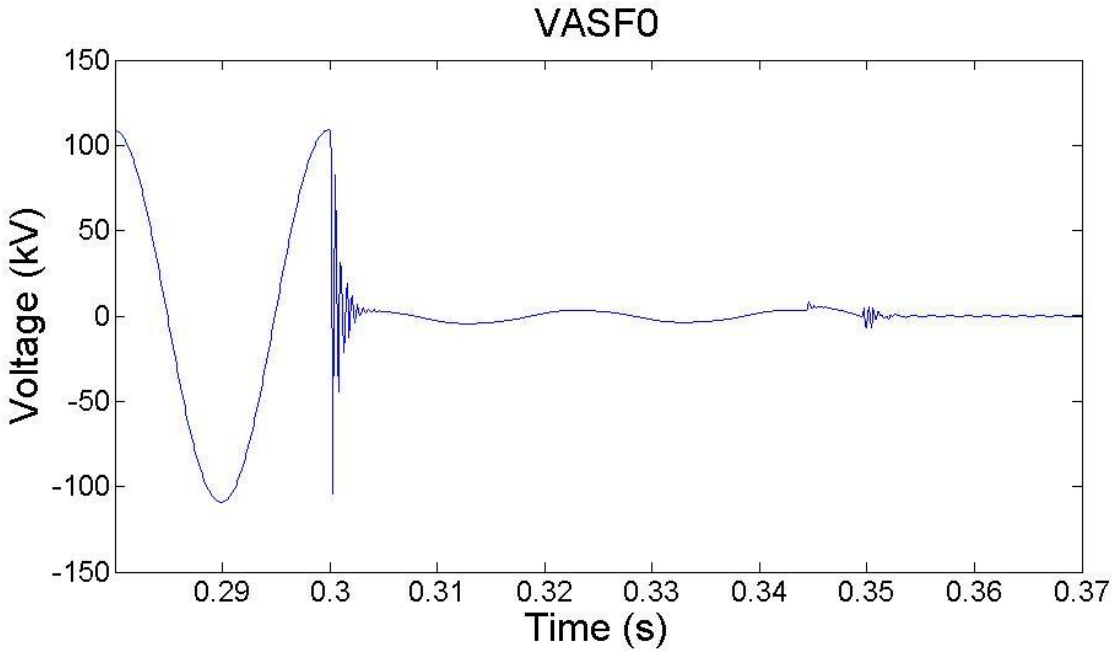


Figure 3.1.2: Faulted waveform for full converter default case

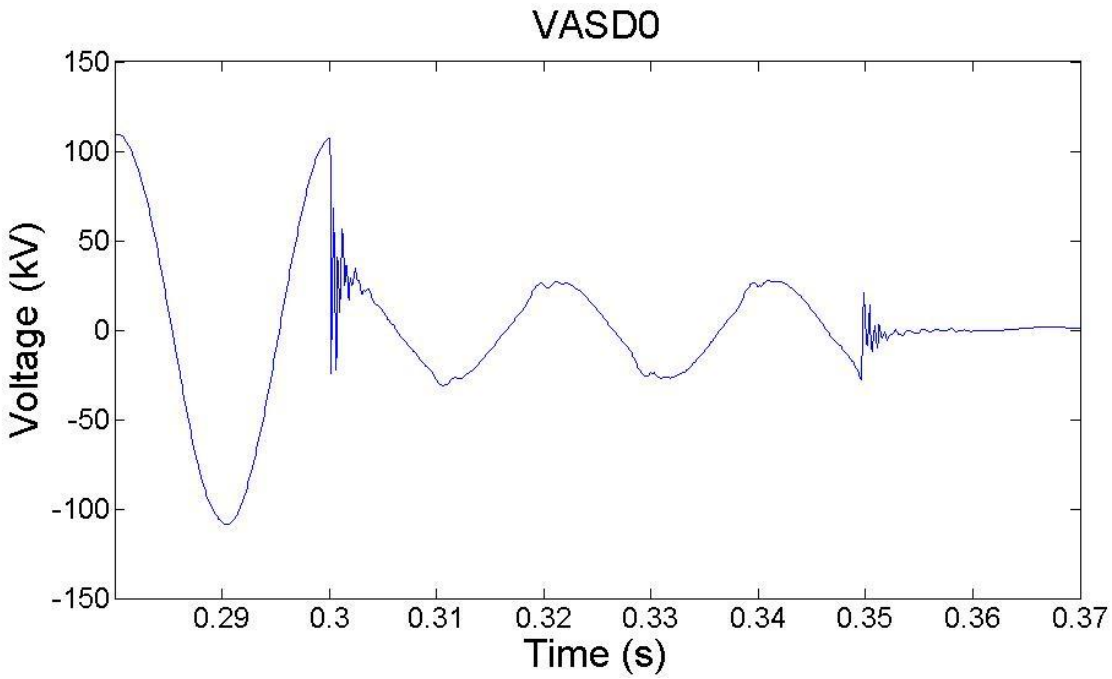


Figure 3.1.3: Faulted waveform for DFIG default case

The DFIG has over twice the intensity at the second peak (caused by the circuit breaker opening), than the other technologies. This is due to the voltage during the fault being considerably higher than the other two technologies. This in turn is likely due to a lack of rotor current control in the DFIG model from the crowbar resistance removal of the control circuitry. In terms of an autoreclosure scheme it would be prudent to take this into account in feature extraction since may affect the secondary arc period.

Across the healthy voltage phases, the FC is slightly different to the conventional case, but the salient features are the same with similar intensities at every frequency band. The DFIG shows much more distortion in the period between the fault and the breaker opening, but the transients due to the circuit breaker are about half the duration of the other cases.

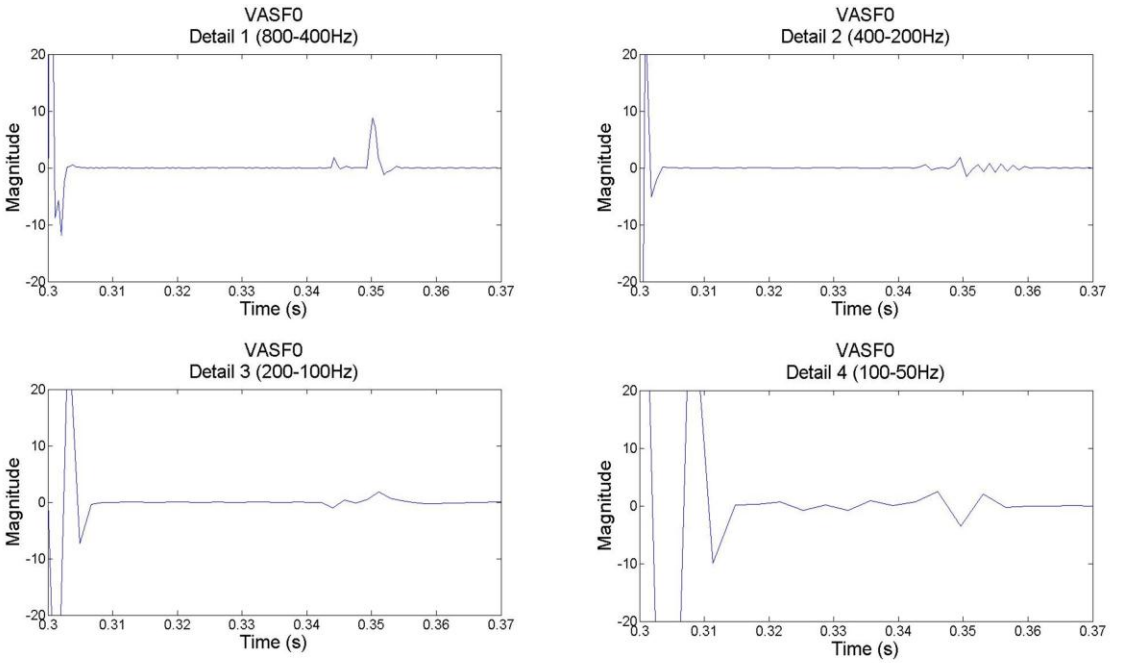


Figure 3.1.4: Wavelet transform details for full converter default case

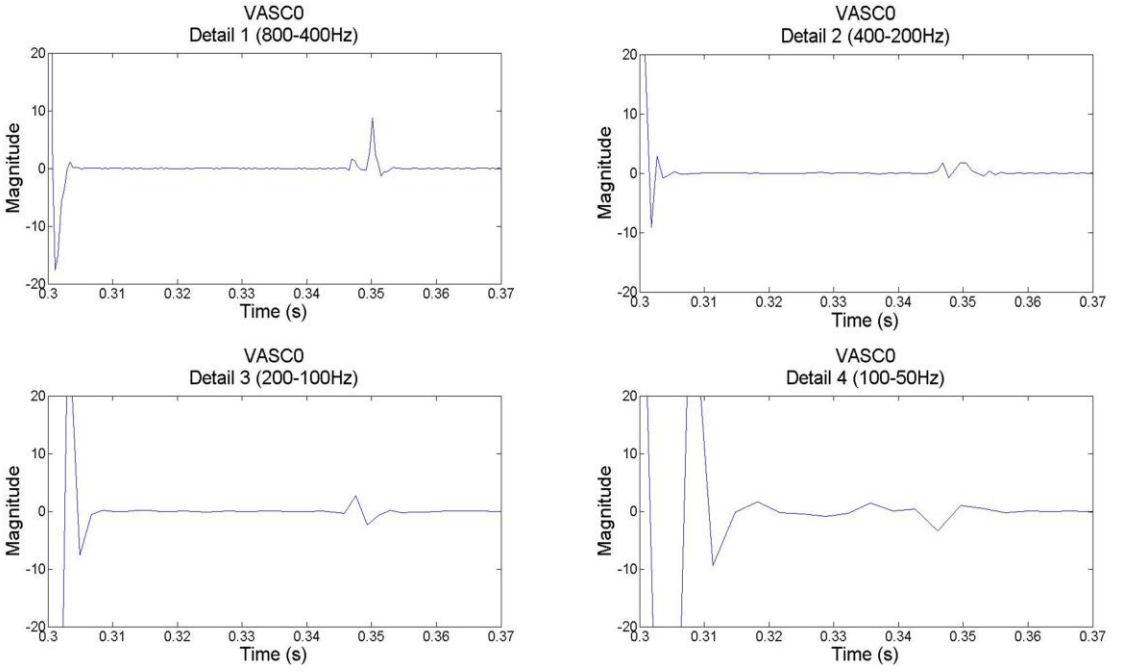


Figure 3.1.5: Wavelet transform details for conventional default case

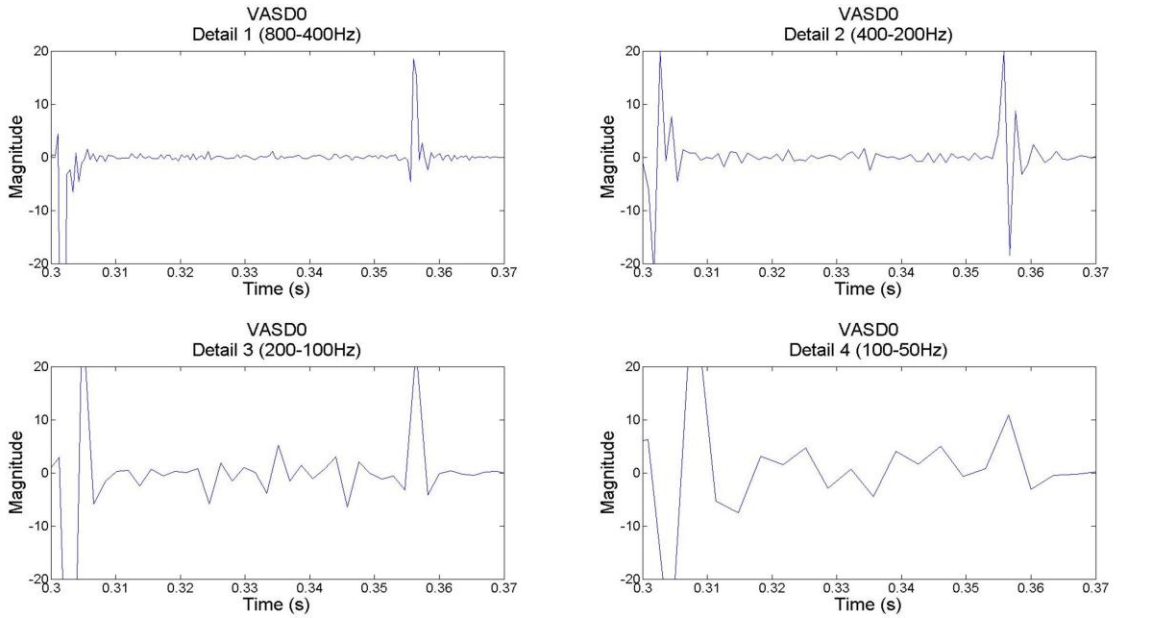


Figure 3.1.6: Wavelet transform details for DFIG default case

II Variance of sending end active power generation

To investigate the effect of generation capacity, the sending end generation capacity was varied from 150 MW to 10.5 MW. (The number of parallel generators were varied rather than the output of existing plant).

In the conventional case, the transients are far smaller for the lower capacity on the faulted phases, but on the healthy phases, there is little difference. Due to the increased power transfer through the line, there is greater phase shift between the currents at each end. This leads to a current zero occurring sooner at the sending end and thus the breaker-opening earlier at this end. The result was a double second spike due to the circuit breaker opening at 340ms. Figure 3.2.1 shows this and can be compared to the default case in figure 3.1.1.

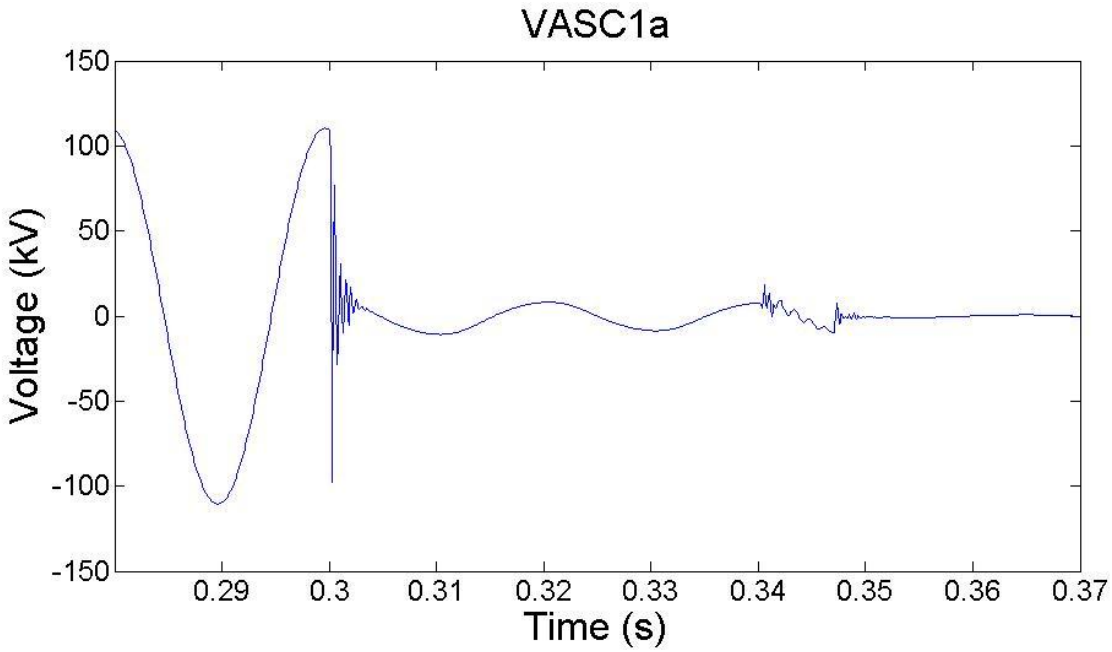


Figure 3.2.1: Faulted phase waveform for conventional 150 MW case showing double transient at 340ms

The FC technology shows very little variation with the size of the wind farm. Although the transient features at each frequency band are slightly different, they share almost identical intensity levels and duration. In contrast, the DFIG shows variation similar to that of the conventional case. When using the DFIG PowerFactory's EMT simulation became unstable and failed at a capacity above 45 MW, so it was only possible to compare the 10 MW against the 45 MW base case. For both the control case and the DFIG there is considerable variation in peaks due to the breaker opening - around twice the magnitude over all frequencies, for the higher generation capacity. The healthy phases show more subtle variation, albeit slightly more than the conventional case.

Surprisingly, the conventional generation shows the greatest variation on the faulted phase. This is possibly due to the lack of power-electronic grid side current or voltage control. From this point of view, any autoreclosing scheme robust enough to deal with varying capacity of conventional generation should be sufficient for wind. In addition, the industry full converter concept is becoming more popular for large turbines [45], which fortunately shows the least variation with wind farm size.

It could be potentially be more useful to study output levels rather than number of turbines. This would show the range power flow that an installed relay would have to cope with for a wind farm as the wind speed varied, rather than just the viability of adaptive autoreclosure on a conceptual level. This would provide further understanding of any issues raised by variability and is suggested as an area for further work. Unfortunately, in this study, changing the output of the turbines is not possible without affecting the validity of the model, due to the complex interaction of generator parameters.

III Variance of short circuit capacity of receiving end.

To investigate this parameter, the short circuit capacity at the receiving end was varied from 35 GVA to 100 MVA. As is expected, the main difference in all generation cases is the level of post fault voltage, which decreases as the local grid becomes weaker. This manifests itself in the wavelet transformation at the 50-100 Hz range as more pronounced features in the 35 GVA case from 300 to 340ms. In the conventional system, the transients are far less prominent at the lower short circuit capacity, especially at the highest frequency band (400-800 Hz). This high frequency feature is also present across the healthy phases. The Full Converter (figure 3.3.1) shows almost exactly the same behaviour as the conventional case with features of very similar duration and intensity.

The DFIG simulation failed at 0.1 GVA. At 0.5 GVA there is considerable distortion in the prefault waveform giving unacceptable power quality. Therefore these results may be disregarded since they do not represent a feasible mode of operation for this technology in the context of this system. However, when the DFIG is connected to the strong grid, prefault operation is stable but there are some interesting differences in the post fault case compared to the other two technologies. The features due to breaker opening at 400 -200 Hz and 200-100 Hz bands are very approximately 10 times larger, even though there is no difference in the signature at 800-400 Hz, as seen in figure 3.3.1 (FC) and 3.3.2 (DFIG). Conversely, the healthy phases contain slightly less high frequency information.

From point of view of feature selection, the Full Converter does not offer any new challenges with the variance of this parameter, but the DFIG would require special

consideration. This is expected since the DFIG is more sensitive to the grid it is connected to because it is not completely decoupled from it. [33]

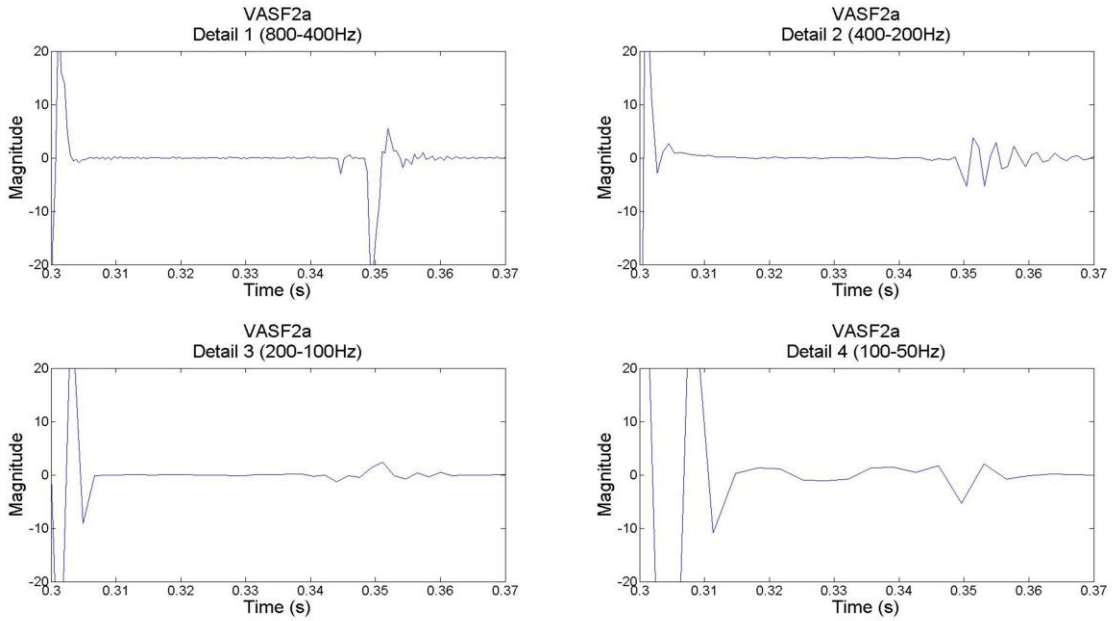


Figure 3.3.1: Wavelet transform details for full converter 35 GVA case

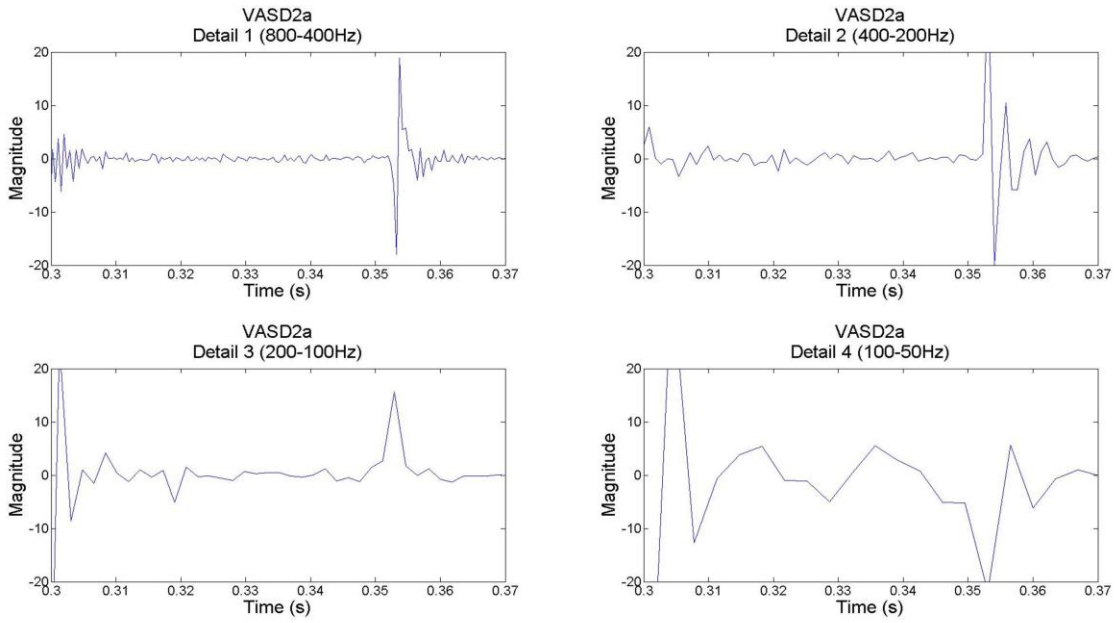


Figure 3.3.2: Wavelet transform details for DFIG 35 GVA case

IV Length of transmission line

The length of the transmission line has clear implications for transient behaviour of the system. Above a certain length, the line must be compensated with FACTS devices, and that brings special considerations for adaptive autoreclosure as previous studies have shown [32]. However, lines beyond 100 km are rare at 132kV network so shunt compensation was not investigated in this study.

In all cases, the transient response to the fault varies dramatically with the changing line length. The full converter and conventional cases show similar variation, as seen in figures 3.4.1 and 3.4.2. At a length of 50 km, the high frequency oscillations caused by the fault do not attenuate by the time the circuit breaker opens, resulting in the two features becoming superimposed after 340ms, and thus difficult to differentiate from each other. At high frequencies, the features are almost identical for these technologies, but at the 200-100 Hz band, the FC model shows a peak of less intensity and shorter duration due to the initial fault. Again, the DFIG shows considerable pre-fault distortion at 50 km, suggesting that this configuration would not be a valid mode of operation and so is omitted from the analysis. When compared to the default line length of 13 km, specified by the UKGDS, the 5 km line has smaller, shorter duration transients across the faulted phase, shown for all cases across figures 3.4.3 – 3.4.5. The healthy phases do not particularly vary with line length. The 5 km DFIG case, shown in figure 3.4.5, is similar to the other two technologies (3.4.3 and 3.4.5) albeit with higher post fault voltage.

The main trend over all phases and all frequencies is much quicker retardation of high frequency transients as line length is reduced. This is due to a smaller amount of trapped charge when the phase became isolated by the circuit breakers. However this was common to all technologies so the implications for adaptive autoreclosure are minimal. It is already well known protective devices must be calibrated for the correct line length regardless of the nature of the local generation.

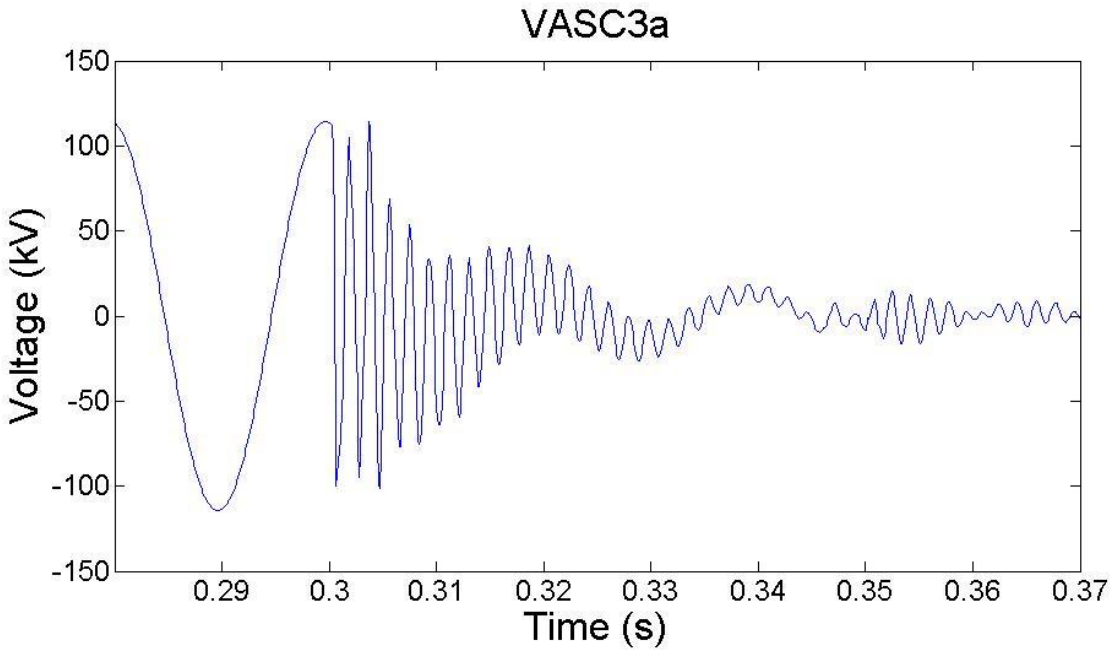


Figure 3.4.1: Conventional generation, 50 km line

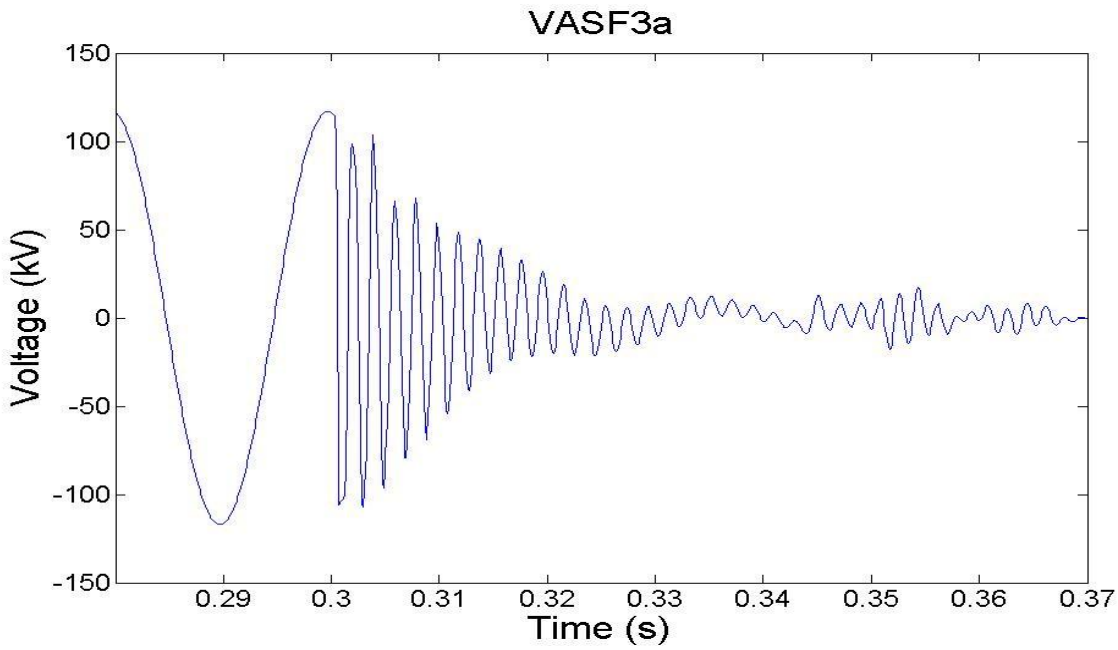


Figure 3.4.2: The full converter case, 50 km line

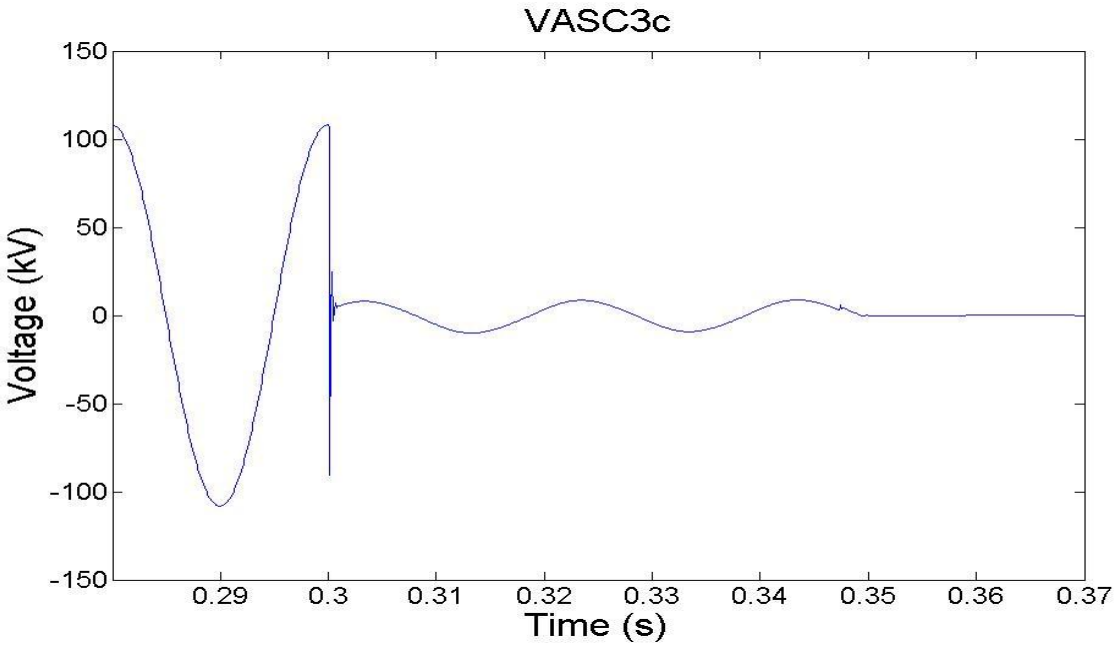


Figure 3.4.3: The conventional case, 5 km line

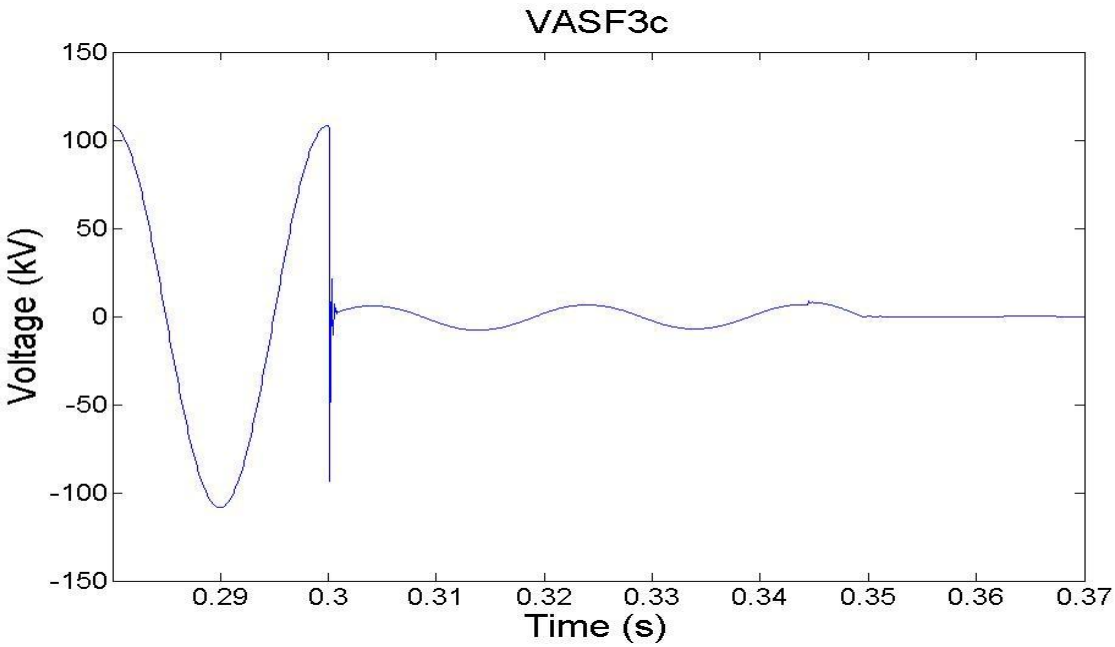


Figure 3.4.4: The full converter case, 5 km line

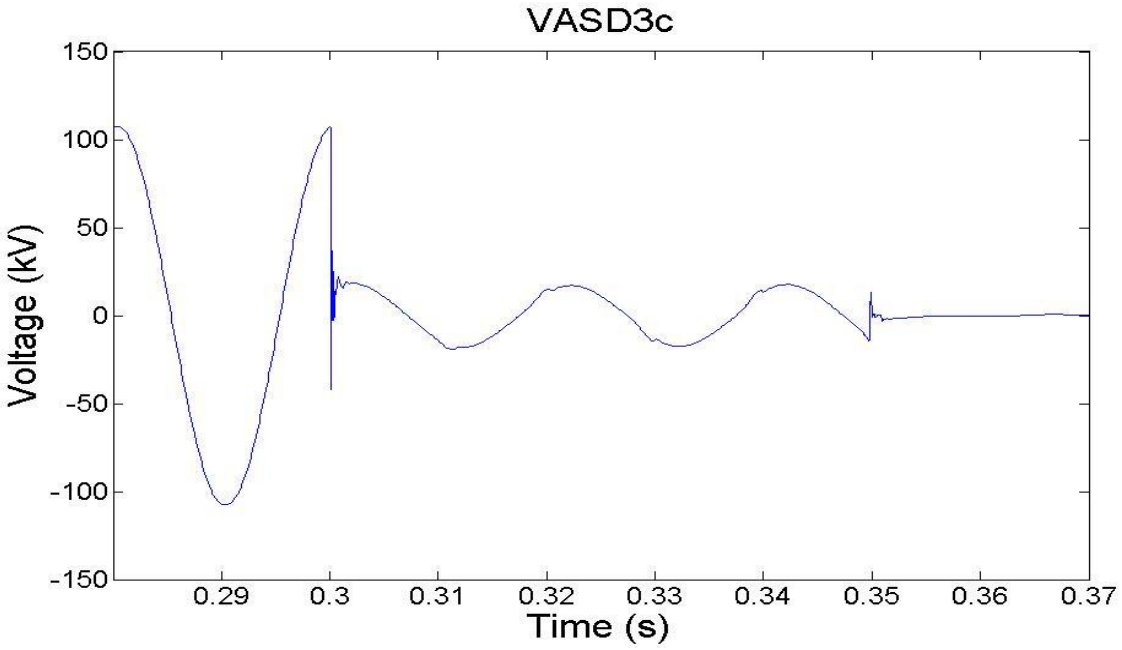


Figure 3.4.5: The DFIG case, 5 km line

V Fault type

To investigate fault types, the three different technologies were compared for a three phase to ground fault, with circuit breakers at both ends of the line changed to trip all three phases.

The DFIG shows very little high frequency activity due to the initial fault, and even less than the conventional case due to the circuit breaker. See figure 3.5.1 for the A phase comparison of conventional case details and figure 3.5.2 for the DFIG details. Similar activity is of course present across all phases since this is a three phase to ground fault. The other phases exhibit less intensity since their waveforms were interrupted at a phase shift of either plus or minus 120 degrees from their maxima.

The FC and conventional cases exhibit very similar behaviour. The FC shows slightly quicker attenuation of the transients due to the initial fault, but conversely has more high frequency information due to the opening of the circuit breakers. This is manifested in the wavelet transform as more high frequency information at the circuit breakers for the FC case, especially in the 400-200Hz band.

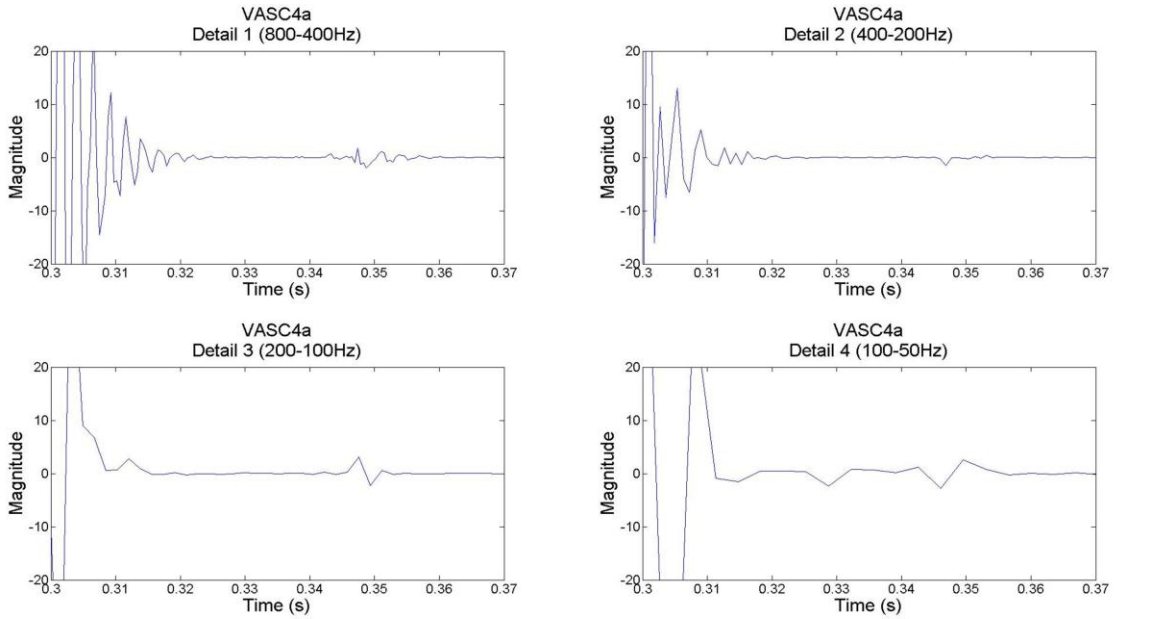


Figure 3.5.1: Wavelet transform for the conventional case for three phase to ground fault

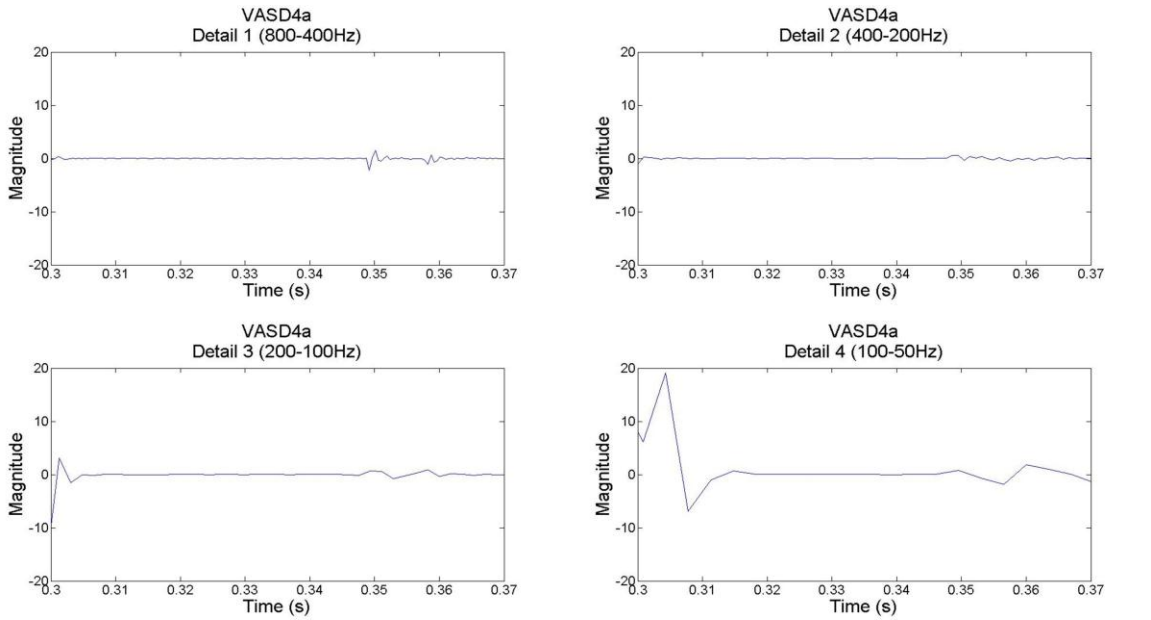


Figure 3.5.2: Wavelet transform for the DFIG case for three phase to ground fault

Given this is the rarest type of fault to occur on a transmission line (less than 4% [46]), it would be more useful to compare other more common fault types. Unfortunately this version of the PowerFactory software does not allow fault analysis for one section of a distributed parameter line model. This limitation can be overcome using two line sections and a 'virtual bus' between them. One phase at this bus may be earthed through a timed single phase switch at the required moment. This was adapted to simulate a three phase to ground fault by connecting through a three phase switch, but could not be extended to other fault types, primary because there is no two phase switch option. In any protection scheme employing single pole and three phase tripping, correct selection of the faulted phase(s) is vital so this is suggested as an area of further study using more suitable software.

VI Location on Line

When the location of the fault is varied from 100 m from the receiving end to 100 m from the sending end, in many respects, the results are similar to the line length variation. For all generating technologies, when the fault is located near the sending end bus, the faulted phase displays no high frequency oscillation post-fault. This is because the amount of line between the measuring bus and fault is minimal - in a way being equivalent to a very short line. Conversely, when the fault is located near the receiving end there is much longer duration high frequency transients due to a longer equivalent line length. Figures 3.6.1 and 3.6.2 demonstrate this contrast, showing the FC case for a fault at 100 m from sending and receiving ends respectively. However, in the case of the transient fault the arc resistance should be removed in a few cycles after the circuit breaker, as the act of opening the breaker clears the fault. This is not considered in this chapter but shown to be very important in later chapters.

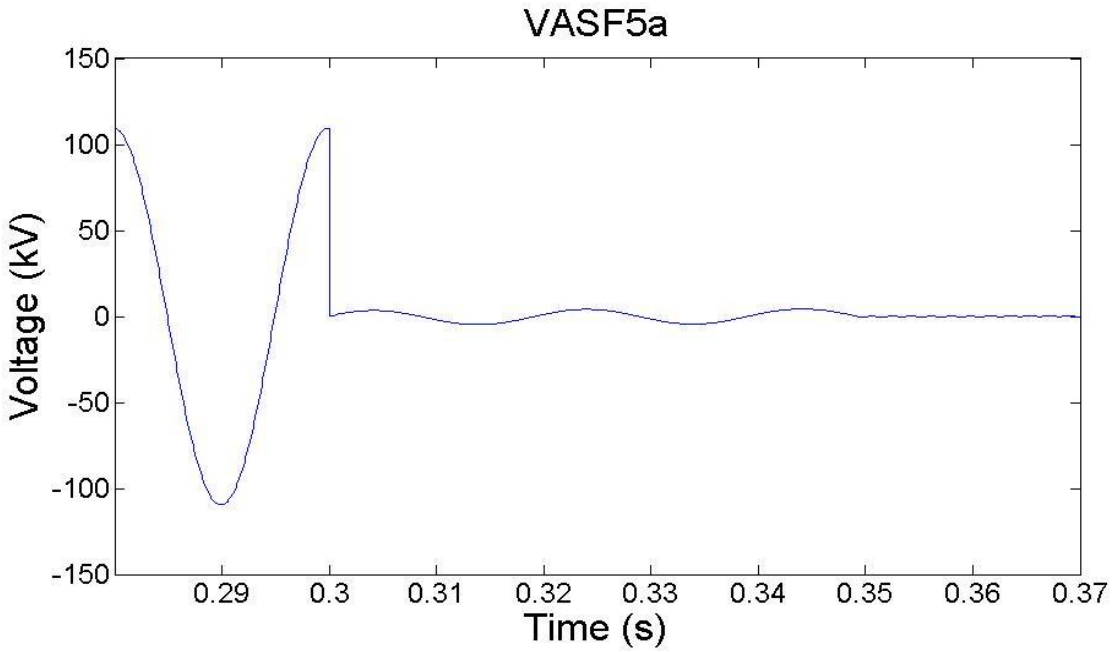


Figure 3.6.1: Full converter case 100 m from sending end (measuring bus)

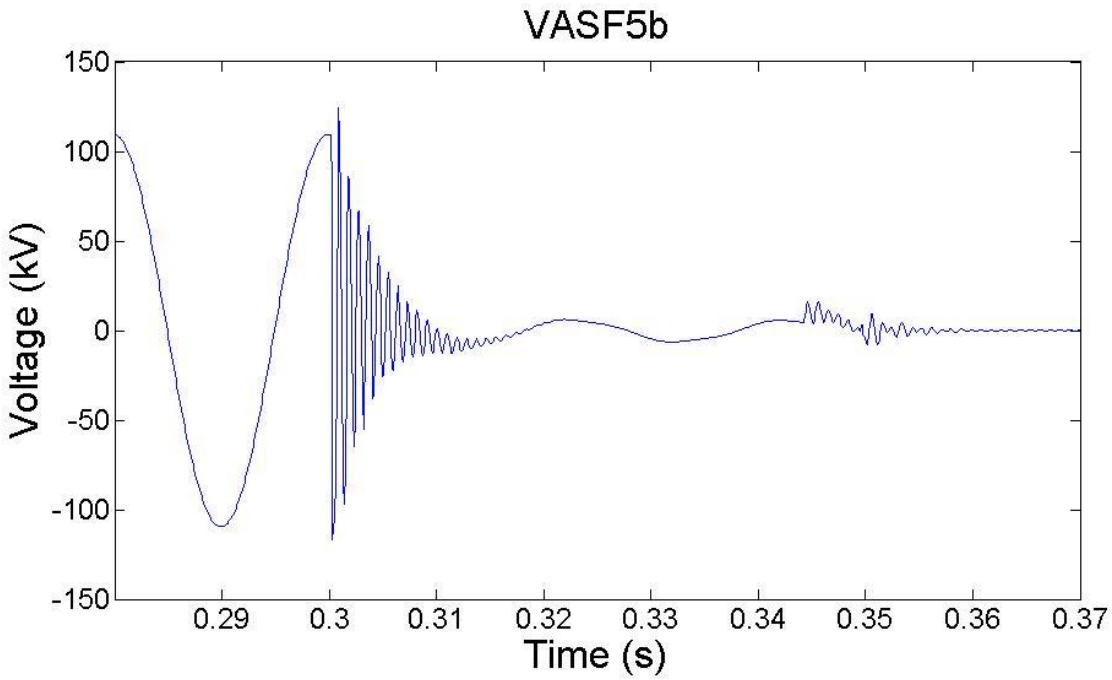


Figure 3.6.2: Full converter case 100 m from receiving end

There are some minor differences between the technologies, but these are far superseded by the variance with fault location. The greatest differences are in the DFIG due to the post-fault noise. Ostensibly, there is a lack of information for diagnostic purposes, on the faulted phase for faults occurring near the measuring bus.

However, this is compensated by the transients on the healthy phases at both events. These persist on the energised phases by virtue of the mutual inductive and capacitive coupling between the faulted phase, see figure 3.6.3.

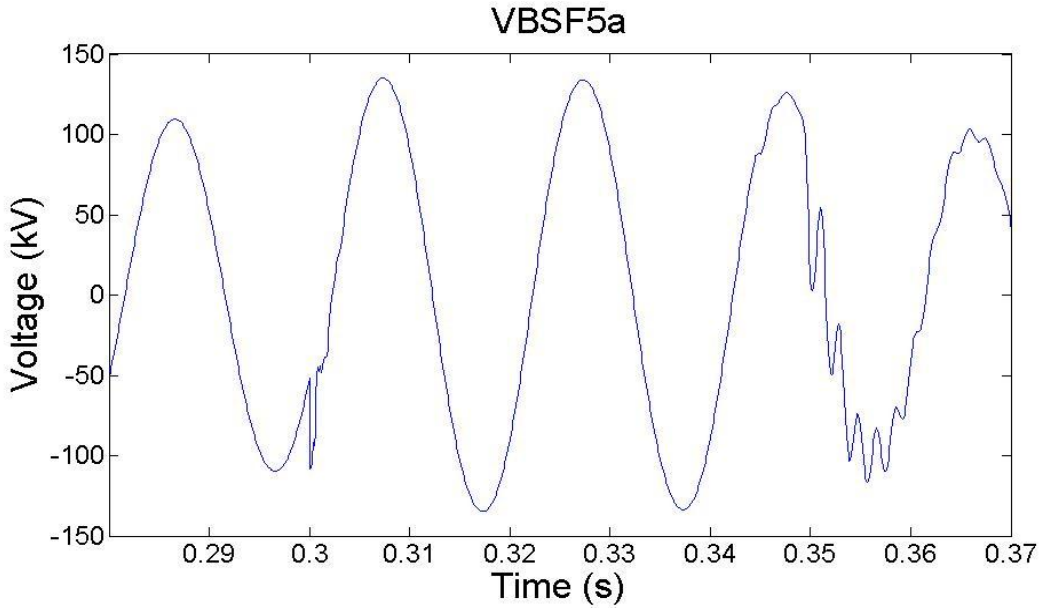


Figure 3.6.3: Phase B, full converter 100m from sending end

Also this lack of information in itself would form part of the pattern recognition when used in conjunction with protection employing AI techniques. Of course, three phase adaptive reclosure would have to make optimal use of the information prior to the circuit breaker operation since in this case, this coupling is not present, as all phases are de-energised. It is not clear how the arc model would affect the waveforms at this stage in the thesis, but the high frequency transients do indicate fault position on line would need to be accounted for.

VII Effect of fault inception point on wave

The default study was set for the fault to occur at a voltage maxima, so for this parameter, the fault timing is varied from 303.33ms to 305ms corresponding to half voltage intensity, and a voltage zero, respectively.

The main results across all the generating technologies is a decrease in high frequency transients as fault level is reduced. The timing of the circuit breakers opening is

unaltered since these remain aligned with the nearest current zero, and so the features from the secondary peak due to the circuit breaker opening are the same as in the default cases.

There are only small differences across the generating technologies, apart from the characteristic high frequency noise associated with the DFIG also observed in the default study.

From the point of view of adaptive autoreclosure, there is very little high frequency information across *all* phases when the fault occurred at a voltage zero. See figure 3.7.1 and 3.7.2 for an example from the FC faulted and healthy phases. High frequency oscillations on the A phase attenuate very quickly, since the starting point is very close to zero displacement. This lack of displacement also meant little information proliferating on the healthy phases. However, existing approaches to adaptive autoreclosing have overcome this since in AdSPAR, the secondary arc period post circuit breaker is most important.

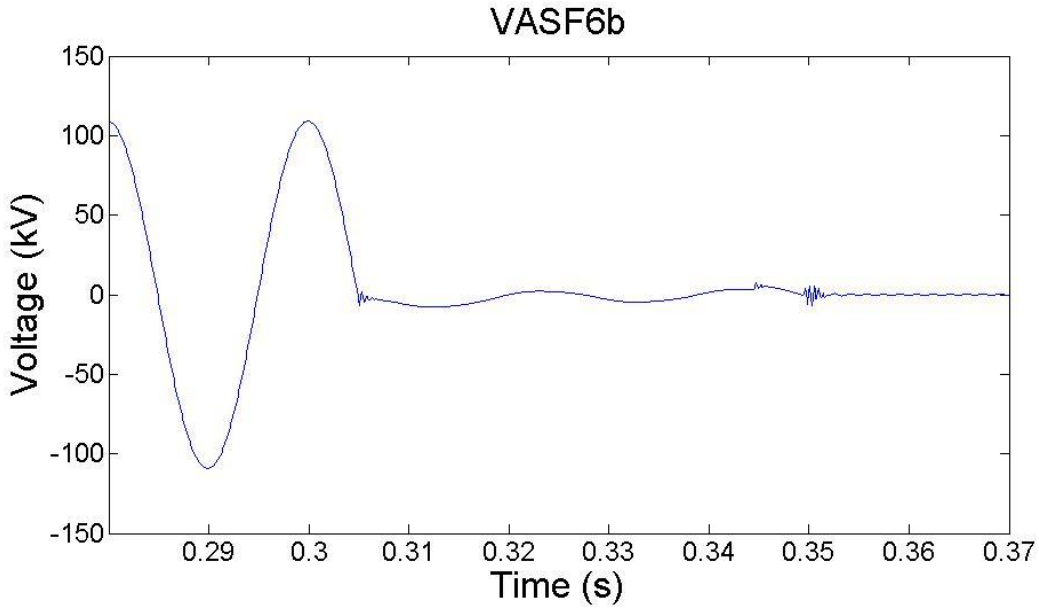
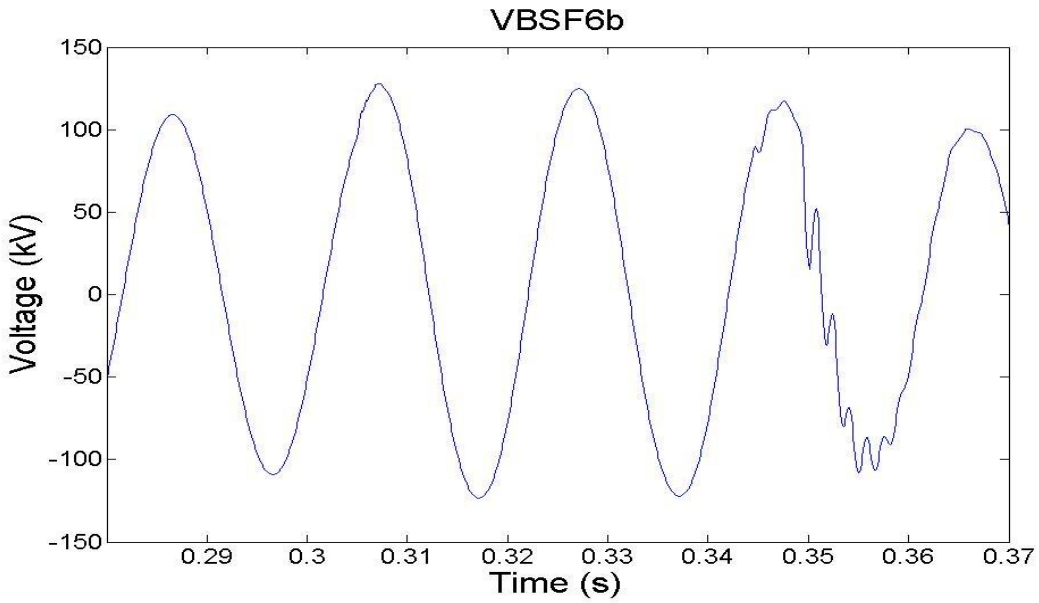


Figure 3.7.1: The full converter case for voltage zero fault inception



Above, Figure 3.7.2: Phase B, the FC case at for voltage zero fault inception

VIII Effect of fault resistance

To investigate the variance of fault resistance, the parameter was varied from zero to 50 Ω. The main task of adaptive autoreclosure schemes is to distinguish between a permanent and transient fault. As discussed in detail in the previous chapter, transient faults exhibit arcing behaviour with a non-linear time-varying resistance. From this point of view, modelling a fault with a fixed resistance is only a valid representation for a permanent fault. However, since the arcing behaviour brings a new level of complexity to the situation, this aspect is best investigated separately, after the effect of wind generation on fixed resistance faults has been established.

The effect of varying the resistance is uniform across all generating technologies. At the high resistance of 50 Ω, the faulted waveforms show slightly quicker attenuation of the initial transient caused by the fault. After the high frequency transients, the fault decays to a steady state with a peak voltage that is about 0.7 p.u. of the peak voltage (much higher than in the low fault resistance case, which is around 0.1 p.u.). In the DFIG case, this fault voltage is about 0.8 p.u. of the pre-fault voltage. These features can be explained by the extra resistance, in addition to the line, between the measuring bus and ground, preventing voltage collapse.

The high frequency effect of the circuit breaker opening is also suppressed at higher frequencies in the case of the high resistance. Across the healthy phases, this effect is

more pronounced, with high frequency transient features decreasing in intensity and duration with increasing fault resistance. This is shown in the FC case healthy phases, in figures 3.8.1 (50 Ω) and 3.8.2 (0 Ω) and the resulting wavelet transform details, figures 3.8.3 and 3.8.4, respectively. This can be attributed to less pronounced coupling as the fault resistance increases due to a smaller post breaker operation voltage collapse. For the solid fault with zero resistance, the effect was much closer to the default case of 2 Ω, suggesting the effect of fault resistance to be fairly linear.

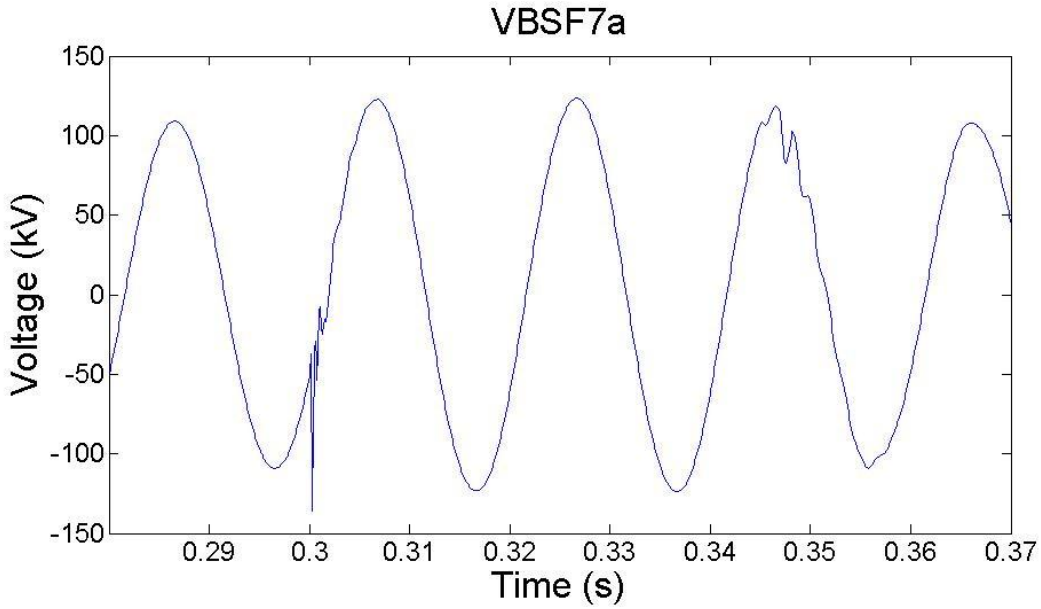


Figure 3.8.1: Phase B for the full converter at 50 Ω fault resistance

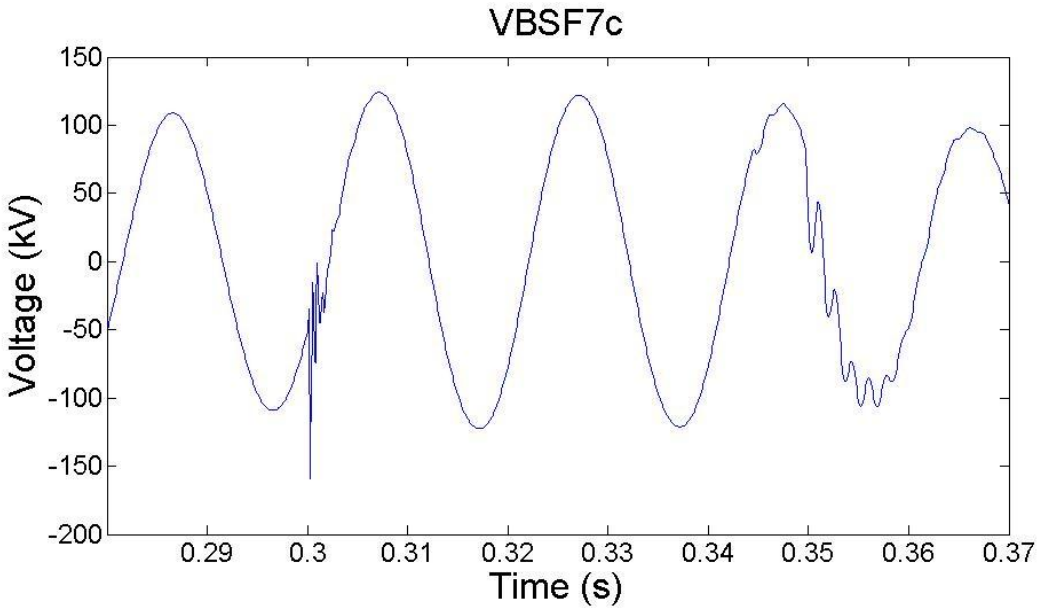


Figure 3.8.2: Healthy phase for the FC at 0 Ω

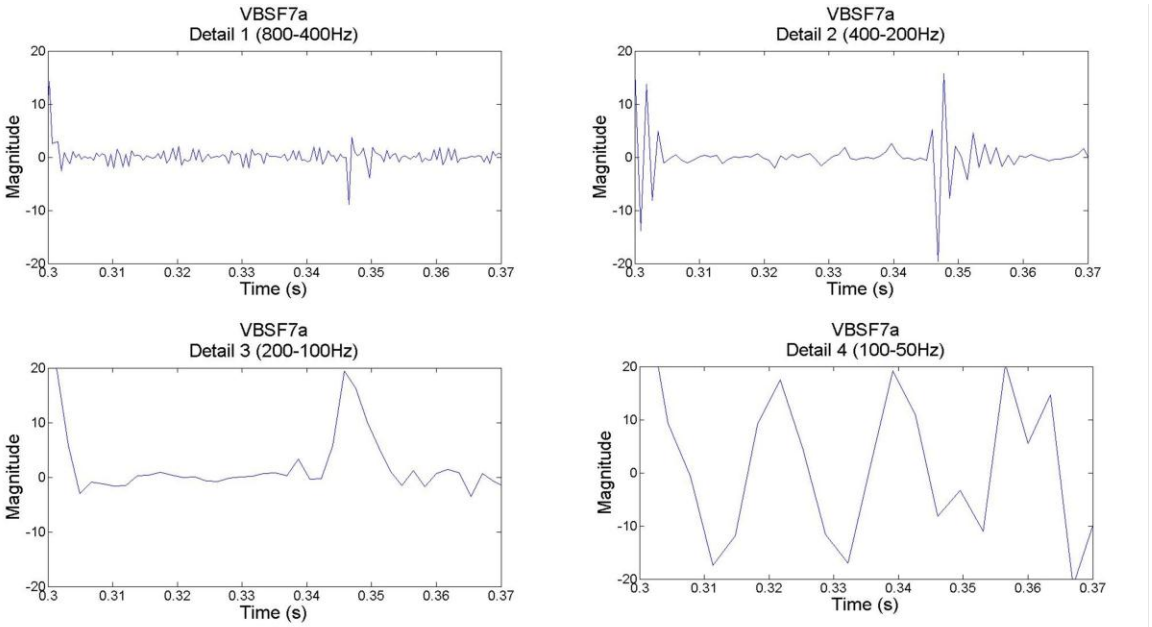


Figure 3.8.3: Wavelet transform details for healthy phase of the FC at 50 Ω

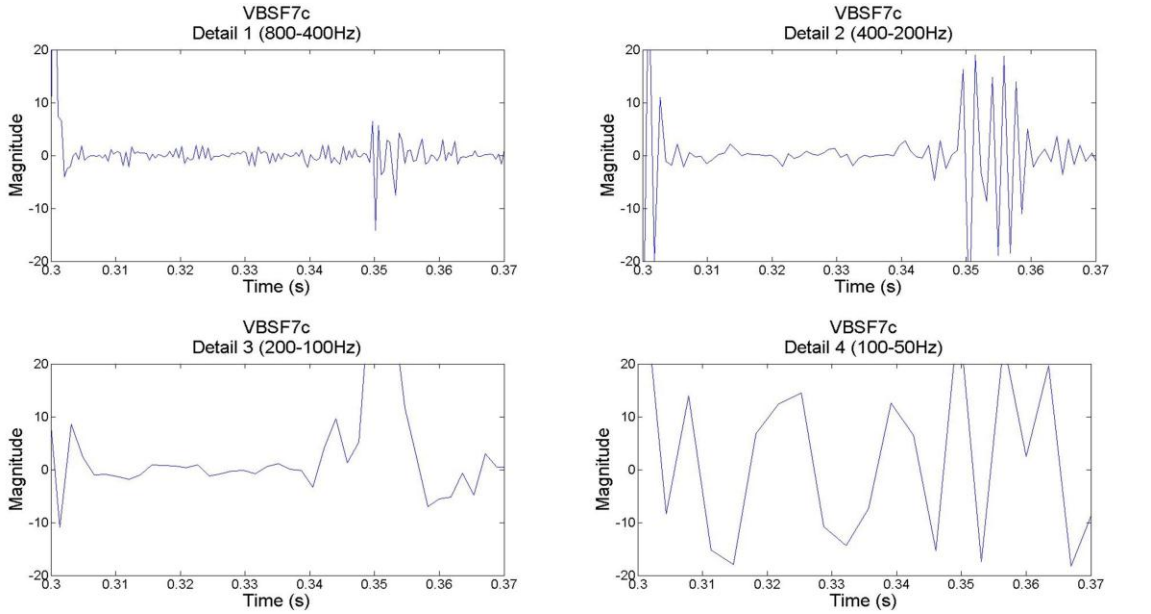


Figure 3.8.4: Wavelet transform details for healthy phase of the FC at 0 Ω

IX Additional stability considerations.

In several of the scenarios using the full converter, the post fault signature on the healthy phases becomes unstable resulting in large over-voltages (in some cases, 200% of the peak-ground voltage), see figure 3.10.1. Comparing this to figure 3.10.2 it can be seen the conventional case does not exhibit this behaviour. The instability appears to be ferroresonance in the transformer connecting the wind farm. This could have implications for any adaptive single pole autoreclosure scheme since the time for which only two phases are able to remain energized would be far less than in the conventional case. With this limitation of an FC-based wind farm, the adSPAR would need to be implemented quickly, or a three-phase trip would become necessary. If the fault fails to clear in the safe time window of two-phase operation, single pole reclosing would be futile because any reclosure attempt would just lead to secondary trips.

However, this behaviour could be due to a computational anomaly in the simulation control loops, rather than the true response of an FC technology. For validation this would require simulation via other software or comparison with real world data, and is thus a suggested area for further study.

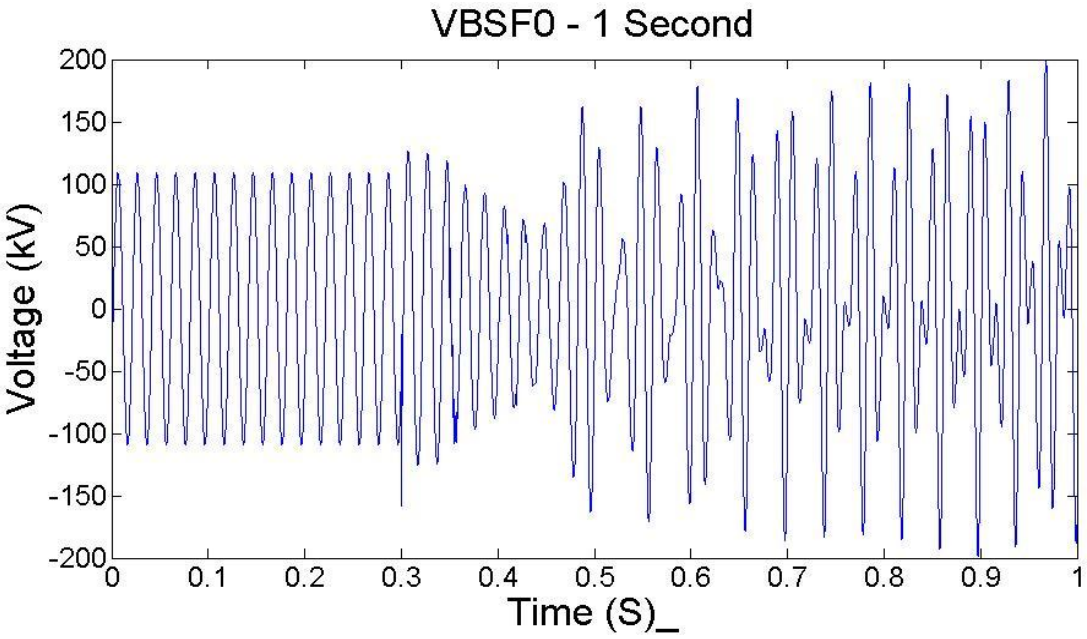


Figure 3.10.1: The B phase over 1 second, FC default case, note overvoltages

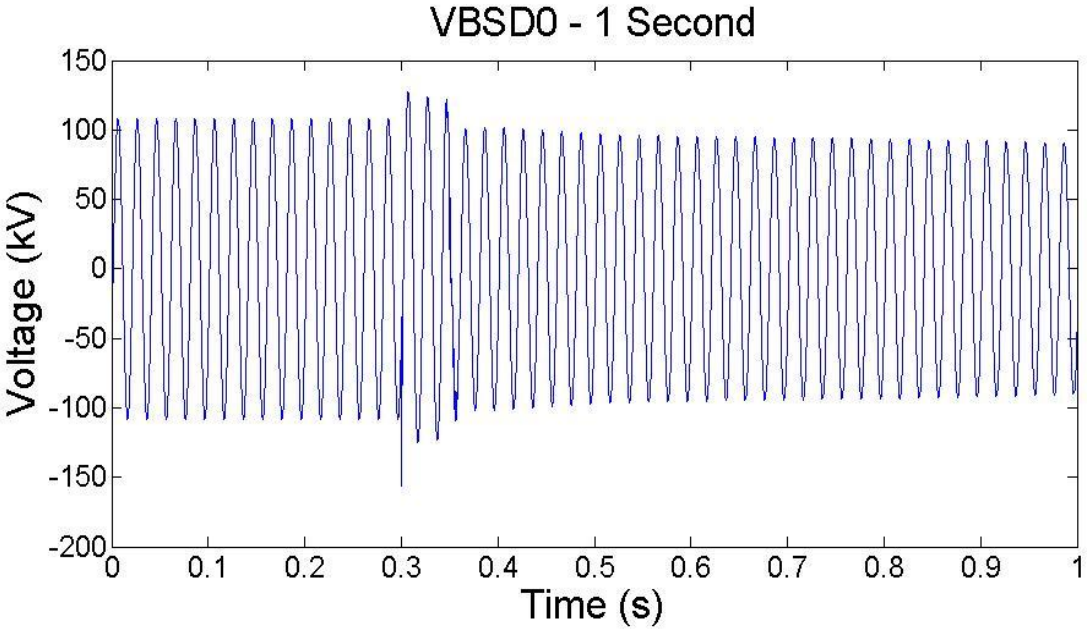


Figure 3.10.2: The B phase over 1 second, DFIG default case, note lack of overvoltages

3-7 Conclusion

As far as generation technology is concerned, the study presented in this chapter has shown that, in general, there are greater differences between the DFIG than the other two cases. This could be down to the greater complexity of the DFIG over the FC model available in PowerFactory. In particular, the DFIG model includes an automatic crowbar resistance across the rotor windings to protect the rotor side converter circuit shortly after the fault. This is common in most DFIG turbines [47], [48]. In practice, full wind turbine transient behaviour is dependent on the individual components and control circuits, but commercial models are not readily available since they are proprietary to the manufacturer. For academic purposes the best approximation for studies of this kind are the generic models like those available in PowerFactory. The trends found in this study would be of benefit for comparison with real world data, where available, for the best validation.

However, the aim of this chapter is not to design an AA scheme specifically for wind turbines, but to investigate the differences and any new considerations may be needed in the design of such a scheme. With this in mind, the following conclusions can be drawn:

1) As mentioned before the DFIG offers the greatest differences. This includes much more post fault distortion, except in the 3 phase balanced fault, where there is actually less than the other technologies. The nature of the grid the DFIG wind warm is connected to is important for AA purposes. Moreover, particular attention should be paid to the control of power electronics during faults, and the protection of the turbine.

2) The differences between the conventional technology are minimal in terms of high frequency transients, although there is a profound difference in how long each technology could continue stable two phase operation, as shown in section 9. This phenomena may be attributable to ferro resonance in the power transformer. This may have a bearing on the suitability of single-phase autoreclosure in lines connected to FC based windfarms, or at least place limitations on its use, because duration of 2 phase operation may be limited.

3) It is also important to consider the effect of the initial protection scheme would have on feature extraction. This is shown in the conventional case and the DFIG on certain parts of the study when the operation of the circuit breakers becomes de-synchronised due to the different local current zeros. Circuit breaker signatures must be accounted for.

4) In all cases, the effect of varying the key parameter has much more significance on the waveforms than the generating technology.

This last point is the most significant finding since it shows existing methods for adaptive autoreclosure are feasible on transmission lines connected to wind farms. It validates the approach, but also highlights the need to consider the type of technology. AI techniques come into their own when used for problems such as this. They are able to recognize trends, extrapolate non-linear relationships and excel at fast pattern recognition. Given these results, the nuances due to different wind generation technologies could be overcome by AI techniques. The calibration of settings may require careful consideration. In particular, if neural networks were used, or other machine learning techniques, the training cases may need to include the generating technology on which the scheme was being deployed.

This chapter is by no means exhaustive. As stated in section II), varying output of existing turbines maybe needed to further investigate issues caused by changing level of power flow. Also for full confidence, the transient response of the wind farm's collector and cabling system, and accompanying substation should be included. In large installations, usually two or more connections are used to introduce redundancy and therefore increased availability. For example, most onshore wind farms have two physical grid connections, each route with double circuit lines. Offshore farms would be connected to the grid with an HV cable that is in some have cases HVDC. In these cases modelling the converter stations is more important than the wind turbines from the perspective of transmission line transients. However, this study does give insight into a worse case scenario. This is because more healthy connections to the grid would damp transients in the faulted line, mitigating the influence of the generating technology.

A fixed fault resistance is a gross oversimplification. A realistic, time-varying arcing resistance is required to model the behaviour of a transient arcing fault. The trends in the study require validation with real world data where available. The methodology used in the chapter may be repeated with an arc model to ascertain any major differences between the generating technologies with this parameter. Ideally, the trends in the study require validation with real world data where available. Failing this, the results must be replicated on other EMT simulation software using a different set of generic models. This last subject is investigated in later chapters.

Chapter 4 – A real time model

4-1 Introduction

The previous chapter assessed the significance of wind turbine technology in the design of adaptive autoreclosing schemes. As mentioned previously, a wind farm connected to a system through a single circuit has the greatest impact on this circuit, and therefore represents the worst case. With this topology, wind generation has the greatest significance since its transient behaviour would not be damped in the system through other network branches. However, it is difficult to meaningfully *quantify* the effect of these technologies. Some kind of quantification is required to answer the questions: Are transients introduced by wind technology high enough to adversely affect autoreclosure? Will they prevent the detection of the arcing signature? The answers, as for most real world engineering questions, may well be 'it depends'. This is down to the variation and complexity in real world system topology. One way to confidently answer these questions would be to deploy AA techniques on a real system with high penetration of wind.

This is not feasible for a number of reasons:

- No utility would permit such a scheme to be tested on a safety critical system for which they are responsible
- Short circuit tests on live power systems are rare since they affect system availability
- Extensive tests are needed both in the training and validation of an AA scheme based on neural networks

The next best alternative is to construct a high fidelity real time model of a real world power system and test the autoreclosing algorithm on this. This would verify that the algorithm is fast enough for real time execution and demonstrate the feasibility of closed loop control. This chapter therefore describes one such real-time, real-world model, using part of the 132kV network in Scotland. The real time digital simulator,

its theoretical background, hardware, software and the network itself are discussed in detail.

4-2 Real time digital simulation

The Nyquist Criterion indicates that the sampling frequency (Fs) must be at least twice the frequency under study. This is because aliasing occurs in the frequency domain at frequencies above Fs/2, so that the frequency spectrum is mirrored about this frequency. Since adaptive autoreclosing relies on the detection of the arcing signature up to 1 kHz, the model must accurately simulate the system frequency response to at least 2 kHz. This means that nodal voltage values and branch currents must be computed in a time step no larger than 500μs. For large systems, this is no trivial task. However, remarkably, the Real Time Digital Simulator (RTDS) is a dedicated hardware unit capable of solving arbitrary system topologies in a timestep of 50μs. This yields a theoretical fidelity of 10 kHz, although RTDS systems specify a lower accuracy limit of 3 kHz.

RTDS background

The underlying computational algorithms of the RTDS are those described by H.W. Dommel in [49]. Notably, the well-known EMTP program, and later the freeware version, ATP, and PowerFactory's EMT program are based on these methods. The trapezoidal rule of integration is used to find an equivalent impedance network for lumped and distributed network components. The entire system matrix is solved in the time domain to give the voltages at each network node. The time domain simulation relies on storing information from the previous time step, of size Δt. Since digital simulation of electromagnetic transients is an important aspect of this thesis, a brief discussion of the techniques outlined in [49] are presented in the foregoing section.

Time domain EMT simulation

The time dependent current flowing through a node is dependent on the voltage drop across k and m.

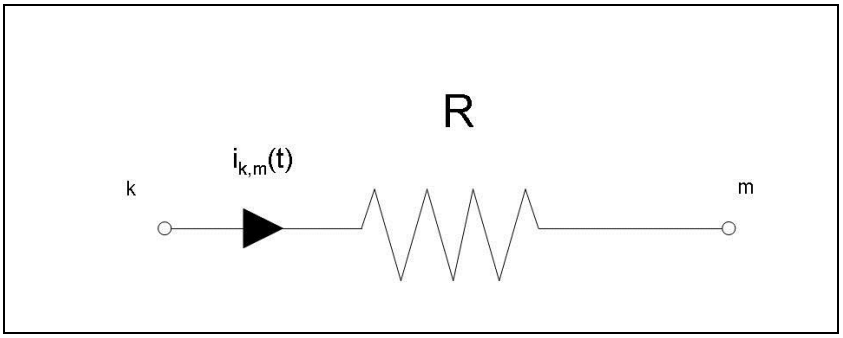


Figure 4.1: A resistor as a time dependent circuit element

For a purely resistive element, shown in figure 4.1, the solution is trivial since it depends on the real form of ohm's law, and no history knowledge of the network is required. Thus equation (4.1) gives the time dependent current through nodes *k* and *m.*

$$i_{k,m}(t) = \frac{1}{R}(v_k(t) - v_m(t)) \qquad (4.1)$$

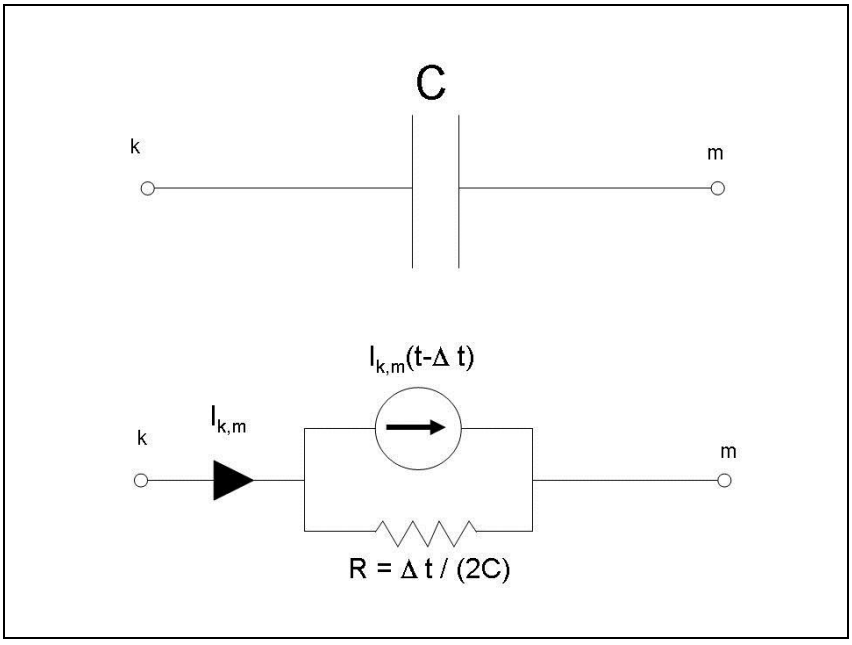


Figure 4.2: A capacitor as a time dependent circuit element

For a lumped capacitance, the equivalent network comprises of a current source in parallel with a resistance. The time dependent potential difference between node k and m is shown by (4.2):

$$v_k(t) - v_m(t) = \frac{1}{C}\int_{t-\Delta t}^{t} i_{k,m}(t)dt + v_k(t-\Delta t) - v_m(t-\Delta t) \quad (4.2)$$

The integral gives the amount of charge transferred by the capacitance during the timestep. When divided by the capacitance, this is equal to the change in potential across nodes k and m. Since knowledge exists of the voltage values at the end of the previous timestep, this yields the new potential drop in the present timestep at time t. The trapezoidal rule of integration is a simple method for finding the approximate definite integral of any continuous function. The area under the graph of $f(x)$ can be approximated to a trapezoid and the area of this shape is then calculated using simple geometry.

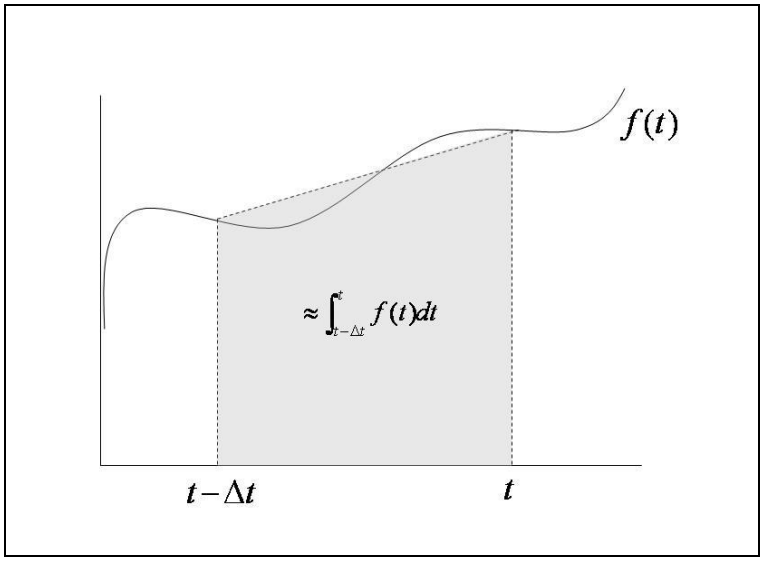


Figure 4.3: Trapezoidal rule of integration

$$\int_{t-\Delta t}^{t} f(t)dt \approx (t-(t-\Delta t))\frac{f(t-\Delta t)+f(t)}{2} \quad (4.3)$$

So as long as the timestep, Δt, is small enough, the integral in equation (4.2) can therefore be calculated with the trapezoidal rule of integration leading to (4.4):

$$i_{k,m}(t) = \left(\frac{2C}{\Delta t}\right)(\mathrm{v_k}(t) - \mathrm{v_m}(t)) + I_{k,m}(t-\Delta t) \quad (4.4)$$

Crucially, the value of the current source is known from the earlier time step and given by (4.5).

$$I_{k,m}(t-\Delta t) = -i_{k,m}(t-\Delta t) - \left(\frac{2C}{\Delta t}\right)(v_k(t-\Delta t) - v_m(t-\Delta t)) \quad (4.5)$$

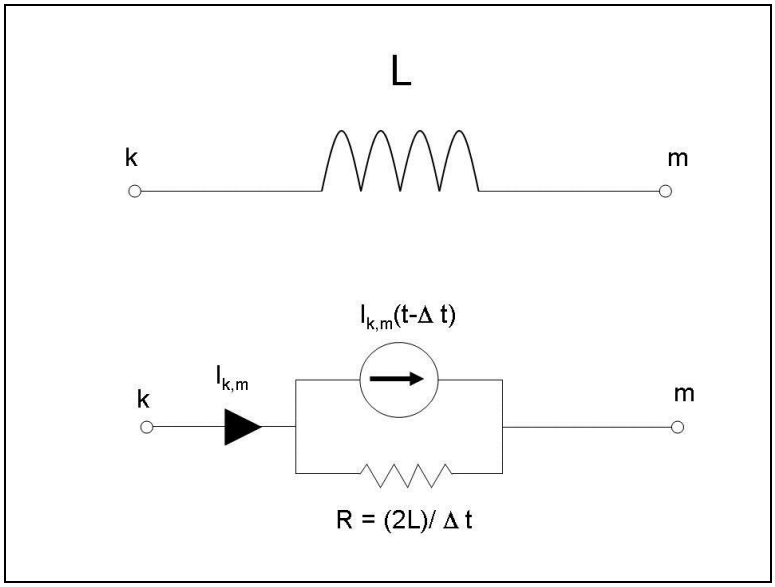


Figure 4.4: An inductor as a time dependent circuit element

For an inductance, shown in figure 4.4, the potential across nodes k and m are equal to the rate of change of current with time through the inductor multiplied by the inductance, shown in equation (4.6):

$$v_k - v_m = L\left(\frac{di_{k,m}}{dt}\right) \qquad (4.6)$$

Integrating both sides over the timestep with respect to t, and rearranging, gives an expression for the current at the new time, equation (4.7):

$$i_{k,m}(t) = i_{k,m}(t-\Delta t) + \frac{1}{L}\int_{t-\Delta t}^{t}(v_k - v_m)dt \qquad (4.7)$$

Again using the trapezoidal rule of integration, this may be evaluated to give equation (4.8).

$$i_{k,m}(t) = \left(\frac{\Delta t}{2L}\right)(v_k(t) - v_m(t)) + I_{k,m}(t-\Delta t) \quad (4.8)$$

And again, the equivalent current source is known from the previous time and shown in equation (4.9).

$$I_{k,m}(t-\Delta t) = i_{k,m}(t-\Delta t) + \left(\frac{\Delta t}{2L}\right)(v_k(t-\Delta t) - v_m(t-\Delta t)) \quad (4.9)$$

When the network involves components with distributed parameters (i.e. L and C are a function of length), such as is the case with transmission lines, the law of characteristics is used to find two equivalent impedance networks, which each comprised of a current source and an impedance. The equivalent circuits for a single-phase line with two terminals shown in figure 4.5.

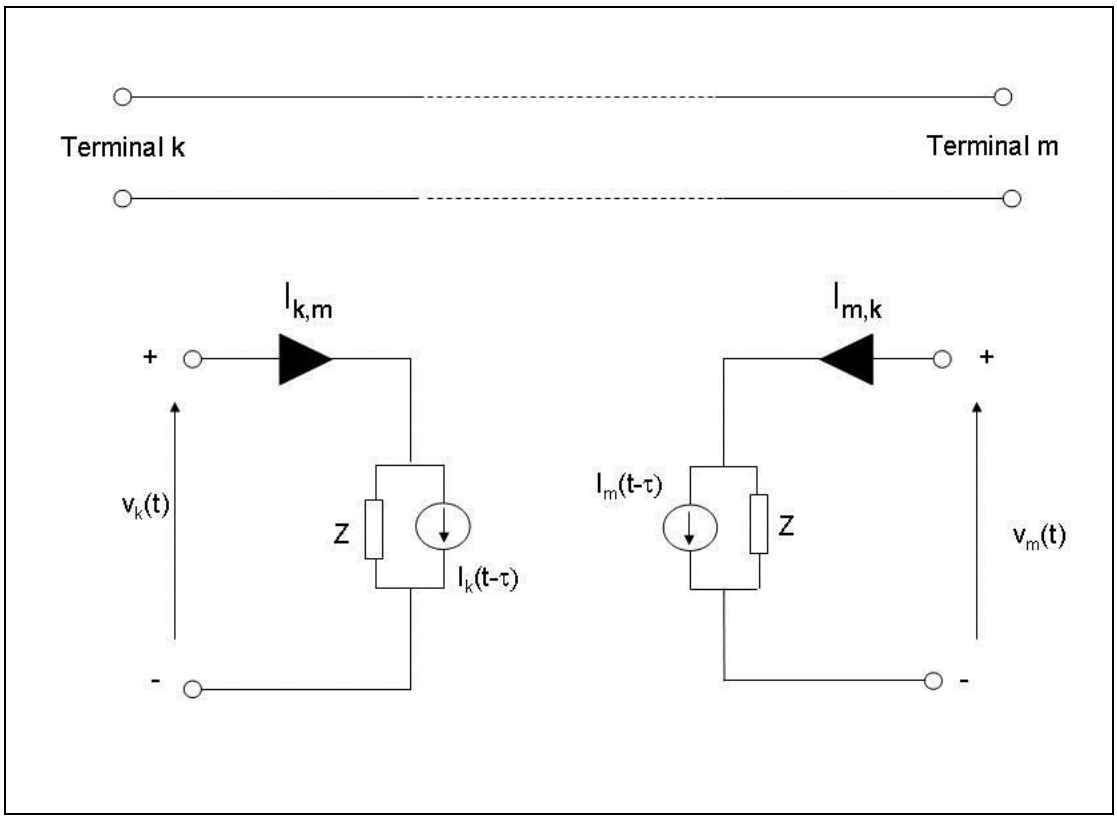


Figure 4.5: Equivalent circuit for a two terminal line with distributed parameters

Assuming there are no resistive losses, voltage and current at point x on a transmission line are related by the partial derivatives (4.10) and (4.11).

$$-\frac{\partial v}{\partial x} = L'\left(\frac{\partial i}{\partial t}\right) \qquad (4.10)$$

$$-\frac{\partial i}{\partial x} = C'\left(\frac{\partial v}{\partial t}\right) \qquad (4.11)$$

Where L' is inductance per unit length and C' is capacitance per unit length. The general solution to these equations is given by (4.12) and (4.13).

$$i(x,t) = f_1(x-ut) + f_2(x+ut) \qquad (4.12)$$

$$v(x,t) = Z \cdot f_1(x-ut) - Z \cdot f_2(x+ut) \qquad (4.13)$$

Where u is the phase velocity, Z is the surge impedance of the line, and $f_1()$ and $f_2()$ are arbitrary functions of (x – ut) and (x + ut).

$$u = \left(\sqrt{\frac{L'}{C'}}\right)^{-1} \qquad (4.14)$$

$$Z = \sqrt{\frac{L'}{C'}} \qquad (4.15)$$

Multiplying equation (4.12) by Z and adding it to (4.13) gives (4.16).

$$v(x,t) + Z \cdot i(x,t) = 2Z \cdot f_1(x-ut) \qquad (4.16)$$

If line is of length d, then the travel time τ, to get from one end to another is (4.17).

$$\tau = \frac{d}{u} = \frac{d}{\sqrt{L'/C'}} \qquad (4.17)$$

In the equation (4.16) the $Z \cdot i(x,t)$ and $v(x,t)$ terms are constant when the expression $(x-ut)$ is also constant. (The expression $(x-ut)$ is one of the 'characteristics' of the differential equation, hence the name for this method.) Consider a hypothetical observer travelling from one end at a velocity *u*. For this observer, $(x-ut)$ and thus $Z \cdot i(x,t)$ and $v(x,t)$ will all be constant. The expression $(x-ut)$ will be the same when they leave at time $(t-\tau)$ as it is when they arrive at time *t*. From this information, the equation (4.18) can written.

$$v_m(t-\tau) + Zi_{m,k}(t-\tau) = v_k(t) + Z\ (-i_{k,m}(t)) \qquad (4.18)$$

That is to say: conditions at node m at time $(t-\tau)$ are the same as conditions at node k at time t. Therefore, the equation, for the current from node k to m, can be written as (4.19).

$$i_{k,m}(t) = -\left(\frac{1}{z}\right)v_k(t) + I_k(t-\tau) \qquad (4.19)$$

And an analogous equation, (4.20), for the current from node m to k.

$$i_{m,k}(t) = -\left(\frac{1}{z}\right)v_m(t) + I_m(t-\tau) \qquad (4.20)$$

In each case, the current sources I_m and I_k are known from the previous point in time $(t-\tau)$, the time when the wave left the opposing port.

$$I_k(t-\tau) = -\left(\frac{1}{z}\right)v_m(t-\tau) - i_{m,k}(t-\tau) \qquad (4.21)$$

$$I_m(t-\tau) = -\left(\frac{1}{z}\right)v_k(t-\tau) - i_{k,m}(t-\tau) \qquad (4.22)$$

Node k only sees conditions at node m through a time delay, τ, through the equivalent current source I_m and vice versa in the case of node m. Resistive losses are accounted

for by adding series resistance at extra nodes at each end of the line, or splitting the line into several places. In the case of three phase lines, mutual coupling must be accounted for by replacing the scalar quantities with vectors that represent each set of the coupled branches. In the phase domain, this produces a series of matrices that have non-zero off-diagonal elements, making the solution far more computationally arduous. The phase quantities are therefore transformed into the modal domain, where there is no coupling, and treated as single phase networks, using the modal travel time and the modal surge impedance.

With the equivalent impedance calculated for every branch and node, one can then build a nodal admittance matrix. Here the node self-impedances are represented by diagonal elements and the inter-node impedances by non-diagonal elements. According to [50], the electrical network problem may then be stated:

"Given a topological nodal matrix A, and a topological branch matrix C, and given the current and voltage source vectors I and V, one must find the instantaneous branch voltages (v) and currents (i) such that Kirchhoff's current and voltage laws hold true and Ohm's law holds true."

Using the techniques outlined in [50], it is possible to build a system of linear equations that relate the nodal current and voltages at time (t) through the constant admittance matrix and the known equivalent current sources, shown by equation (4.23).

$$[Y][v(t)] = [i(t)] - [I] \qquad (4.23)$$

The network may then be further subdivided into subsets of nodes with known voltages (i.e. voltage sources) and those with unknown voltages. The resulting sub matrices are rearranged so that the column vector of unknown voltages can be solved in each timestep. Provided the timestep is fixed, $[Y]$ also remains constant, greatly increasing the speed of the solution.

It is important at this point to remark upon the difference between the previous time step, Δt and τ, the travel time from one terminal of the distributed parameter component to the other. In the case of lumped parameters, the history state at t - Δt is required, and with distributed parameters, the history state at t - τ. Moreover, in the case of distributed parameters, the simulation timestep must be smaller than the travel time of a distributed parameter line. In the case of overhead lines, the modal velocity u is just below the speed of light, approximately $3 \times 10^8 ms^{-1}$. In the case of the RTDS, real time simulation takes place at a base timestep of 50μs. Given that the speed of light travels 15 km in this time, no line section below 15 km can be represented with a travelling wave model.

Transmission line parameters are also frequency dependent as well as distance dependent. This does not matter for steady state (50 or 60 Hz) analysis. However, when there is a step change in system conditions, in the form of a fault or a switch, the resulting voltage and current oscillations occur at many frequencies. The work of Semlyen [51] was the first to accurately predict the transient behaviour of transmission lines. Using convolution methods to transfer results into the time domain is possible but computationally intensive. The Semlyen method overcomes this problem using two exponentials to approximate the unit step response function needed for the computation of line transients, and thus simplifying the integral. However, the approximations used result in large inaccuracies, even in transposed lines. A valuable contribution to the solution was made by Marti in 1981 through his PhD work [52]. In previous methods, the convolution at each time step had been modelled by the superposition of the forward wave and the returning wave. The contribution of each wave was evaluated by a "weighting" function which was analogous to the historical contribution of voltages at each end of the line. Marti's contribution was to introduce a receiving network whose weighting function cancelled out much of the contribution of the forward weighting function, greatly reducing the complexity of the convolution integral, and in turn streamlining the computation time. This approach results in an algorithm that is only around 30% more computationally expensive than non- frequency dependent models. The limitation of this model is that it assumes a constant transformation matrix in the decoupling of the line phases, which is in reality frequency dependent. This is particularly erroneous for

untransposed lines and underground cables, and can lead to low frequency errors and asymptotic voltages in some cases. For cases involving a trapped charge, the low frequency instability can lead to misleading behaviour of the DC component. However, since Marti's methods (applied via ATP software) have been successfully used to develop Fitton et al's AA technique, they can be used with confidence in this work. RTDS systems have recently developed a phase domain model, whereby the admittance matrix is solved directly in the phase domain. The phase domain model is very resource intensive, so may only be used sparingly in a large simulation.

In the case of the RTDS, the small timestep is achieved in *real time* through advanced parallel processing techniques. Several processors mounted on racks solve individual models and communicate with each other sub-timestep so that the entire network solution can be solved in a base frame rate of 50μs. The user is able to draft power systems of arbitrary complexity, the size of which are limited only by the amount of processing power available. These can then be run in real time, with the user interacting through control actions, and observing the response of the power system.

4-3 RTDS Hardware

Whilst this thesis endeavours to be commercially neutral, and thus in no way favour RTDS technologies over its competitors, some in depth discussion of RTDS hardware is required to describe how the primary system model was realised. This section is not intended as an advertisement.

The main computational horsepower of the RTDS consists of two types of processing units, 3PC and GPC cards. The older 3PC card houses three DSP processors (ADSP−21062 SHARC Processors), each with a clock speed of around 40 MHz. However, the highly parallelised architecture means that some software applications can operate at up 120 MFLOPS. The RTDS system used in this work contains ten 3PC cards, five in each rack. The newer Giga processor card (GPC) has two on-board 1GHz IBM PPC750GX PowerPC (RISC) processors operating at a floating point rate of 1GFLOP, according to [53]. A single GPC processor is capable of solving a network solution of up to 54 nodes, or 17 busses in a three phase network. The most

onerous computational tasks are handled by these units. These include small time-step modelling, the highest fidelity transmission lines and the network solution itself. In the RTDS system used, there were a total of four GPC cards, two in each rack.

It is of course vitally important in the RTDS simulation that the base timestep of 50μs is adhered to for the duration. Communication between cards is achieved via the RTDS's backplane, a bus coordinated by the workstation interface card (WIC). This ensures that all relevant information is passed between processors sub-timestep. The WIC card also handles LAN based user communications between the RTDS and the workstation. This allows user interaction in runtime and updates real time displays such as plots and meters. LAN based communication is not fast enough to keep up with the timestep, so informational feedback to the user occurs in pseudo-real time. Information is streamed over LAN in packets and used to update runtime. Plots and meters can be refreshed manually or are updated shortly after a user control action has been initiated. Further information regarding RTDS hardware can be found in [53].

4-4 132kV Network

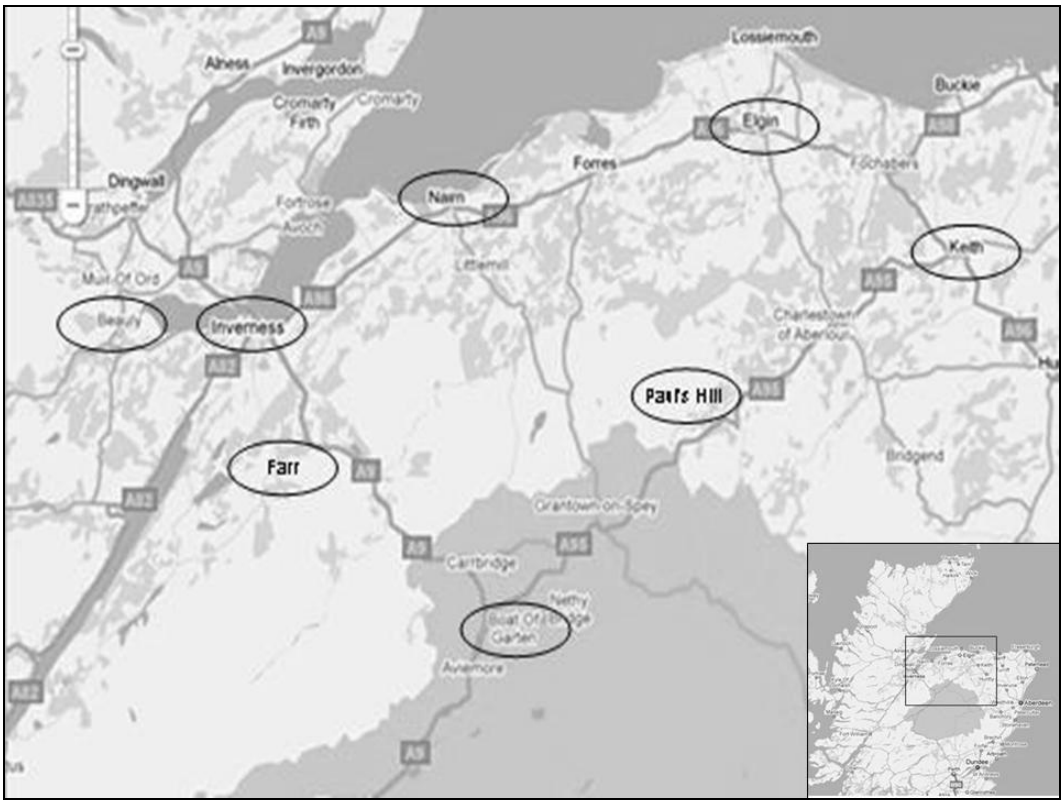


Figure 4.6: Map of network modelled

The chosen section of network is situated in the Scottish Highlands and covers an area of approximately 1300 km^2. Figure 4.6 shows a map of the area with the locations of the key substations. The network comprises of eight busses arranged in a ring topology shown in figure 4.7. The northern branch, approximately 90 km, runs from Beauly to Keith supplying the load centres Inverness Nairn, Elgin and Keith. A southern 50 km branch from Beauly supplies Boat of Garten, collecting power on route from the Farr wind farm. A second 50 km branch runs from Boat of Garten to Keith, via Paul's Hill wind farm. This last branch is a single circuit line.

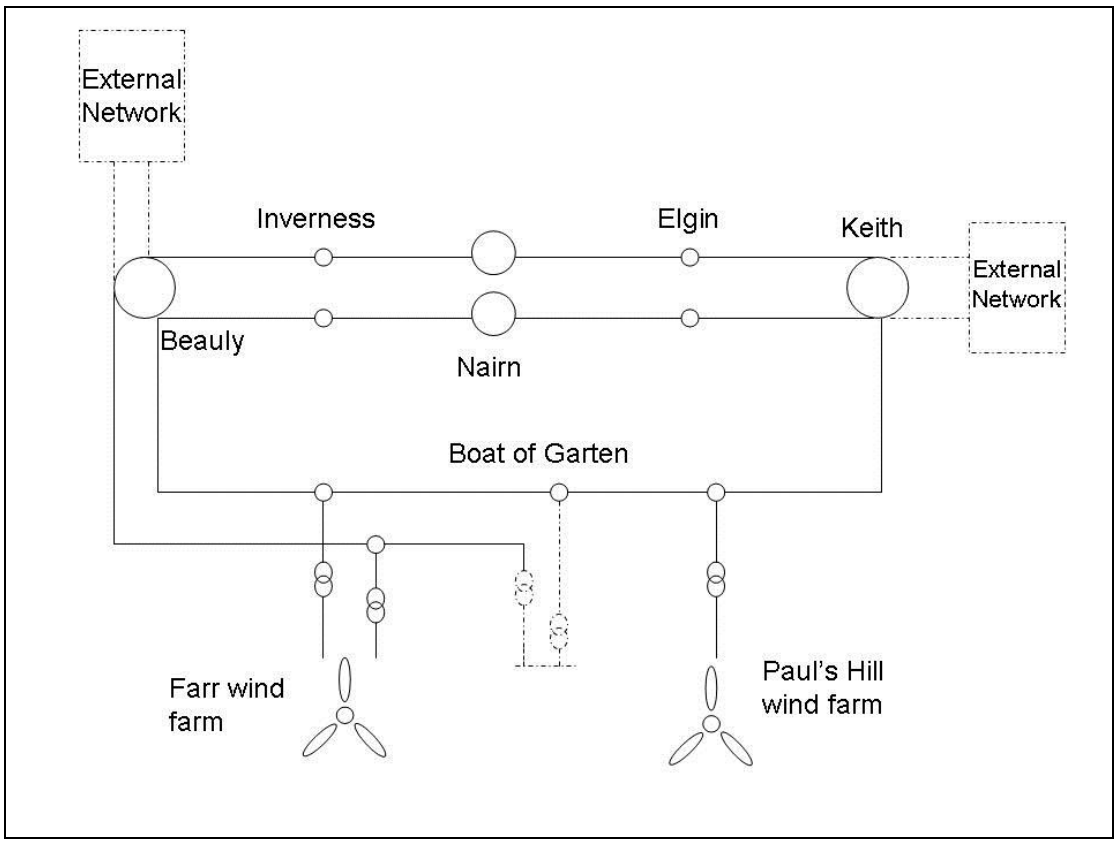


Figure 4.7: Diagram of network modelled

This is suitable since it fulfils several criteria:

- It includes two large wind farms
- Both wind farms use DFIG technology, of the same manufacturer (Bonus 2.3MW)

- It includes lines that are long enough to represent with the RTDS travelling wave OHL model
- The entire network ring is at 132kV, obviating the need for simulation of power transformers in the network
- The extent of wind farm transients can be observed on distant lines as well as nearby lines
- Double circuit lines exhibit inter-circuit mutual coupling, giving rise to the possibility of three phase adaptive autoreclosing.

Most model data was obtained from National Grid's 7-year statement [54]. However, some more detailed information was required for some aspects of the simulation. This included in the conductor layout and configuration in the overhead lines. This was kindly supplied by Scottish and Southern energy and National Grid, but is not currently published material so cannot be referenced.

Source models

The presence of the external network at Beauly and Keith were represented by two voltage sources behind an equivalent short circuit impedance, to represent the short circuit capacity at that bus, quoted in the 7-year statement. These quantities are related by equation (4.24).

$$S.C.C. = \frac{V^2}{Z} \qquad (4.24)$$

Where in (4.24), S.C.C. is the short circuit capacity in MVA, V is the RMS line-to-line voltage, in kV and Z is the short circuit impedance. Rearranging gives equation (4.25).

$$Z = \frac{V^2}{S.C.C} \qquad (4.25)$$

Thus the parameters in table 4.1 where used:

Table 4.1: Model source parameters

Source	Capacity/ GVA	Impedance /ohms	X/R ratio	Source Angle /degrees
Beauly	10.199	1.708	13.5	85.8
Keith	6.198	2.811	13.5	85.8

X/R ratio here is not available from [54] so an average was taken from [55], where X/R at distribution is quoted between 2-25.

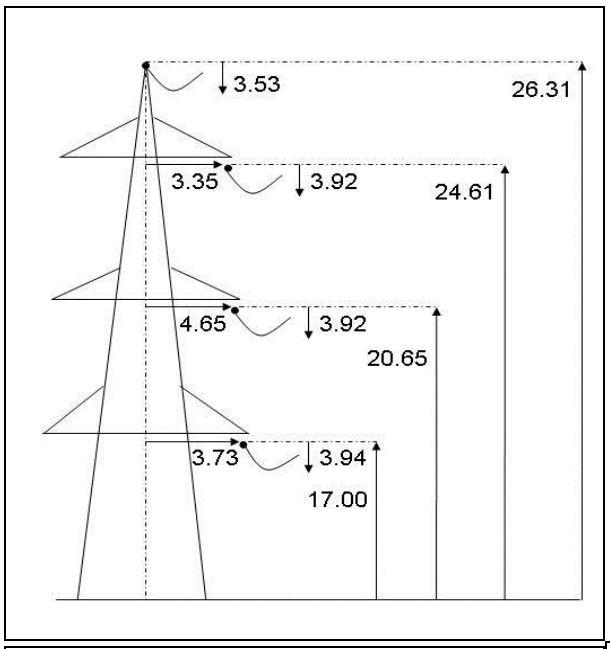
For the load flow component of the RSCAD system, values from National Grid's 7-year statement 'peak winter load flow' were taken. This was necessary for the initialisation of the simulation. As the simulation runs in real time, load flow settles to a steady state, which in the model was very close to initial values specified.

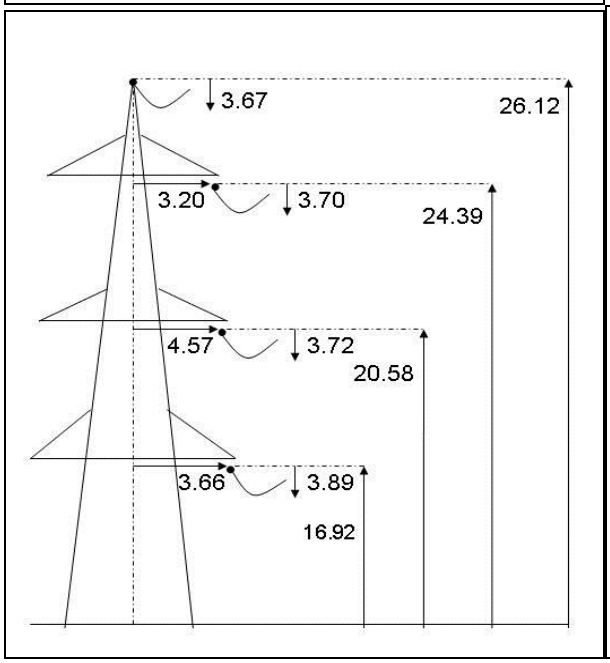
Line modelling

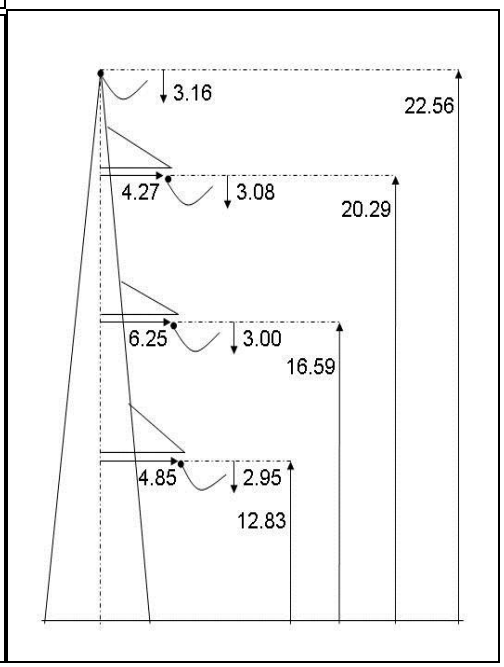
One of the most important aspects of the model is high fidelity transmission line modelling. As mentioned in section 4.2, this is due to the distributed nature of the capacitances and inductances, and the frequency dependence of these parameters. In three phase systems, the conductors are mutually coupled. On the UK's system, there has been a consistent approach to double circuit transmission routes, i.e. double circuit towers with three phase conductors either side and a shared neutral wire. This configuration means that, not only is there mutual coupling between the phases, but also coupling between the circuits. This inter-circuit coupling leads to a secondary arc even following the opening of a three-phase circuit breaker, provided the other circuit is energised post fault [26]. The presence of secondary arcing yields the possibility of deploying adaptive autoreclosing on systems with three phase breakers, and is investigated later in this thesis.

RSCAD offers several line models, the most accurate of which is the frequency dependent, phase domain model for six mutually coupled conductors spread over two

circuits. Referring back to section 4.2, this model overcomes inaccuracies in the transformation matrix by solving the admittance matrix directly in the phase domain. Due to the off-diagonal matrix elements, this highly computationally intensive model requires an entire GPC processor. Consequently, it was only possible to represent one line with this model with the RTDS resources available. Therefore, the transmission line on which the adaptive autoreclosing algorithm was deployed was the only line to use this model. This was designated the ‘high fidelity area’. Short circuit tests require two line sections, since the line must be split at the fault point to represent the current path to ground. The geometric layouts shown in figures 4.8 - 4.10 were used. All measurements are in meters and the sag at mid span (between successive towers) is measured by the vertical displacement from the height of the conductor at the tower. The average height of the conductors is calculated from their height at the tower minus 2/3 of their sag at mid span.







Figures anticlockwise from top, (4.8), line type 1, (4.9) line type 2 and (4.10) line type 3. Figures show the geometric conductor configuration for twin circuit towers (4.10 single circuit) with shared earth wire. Vertical and horizontal displacement, and mid span sag also noted. Measurements in metres.

The Type 1 line, shown in figure 4.8, is used on the Beauly to Boat of Garten via Farr wind farm route. This is where the autoreclosing relay is deployed, in the section spanning Farr to Beauly at the Farr bus. The Type 2 line configuration, figure 4.9, was used on all other lines, except the single circuit line section from Boat of Garten to Keith via Paul's Hill wind farm.

The lines outside the 'high fidelity area' (figure 4.11 and figure 4.12) were represented with RSCAD's frequency dependent modal domain model. This is a faster yet slightly less accurate model since the transformation matrix used to convert between phase and modal parameters is assumed frequency independent. The line

conductors were all a single conductor bundle per phase of the lynx type, with 0.967 cm radius. In all cases, the ground resistivity was taken to be 100 $\Omega.m^{-1}$.

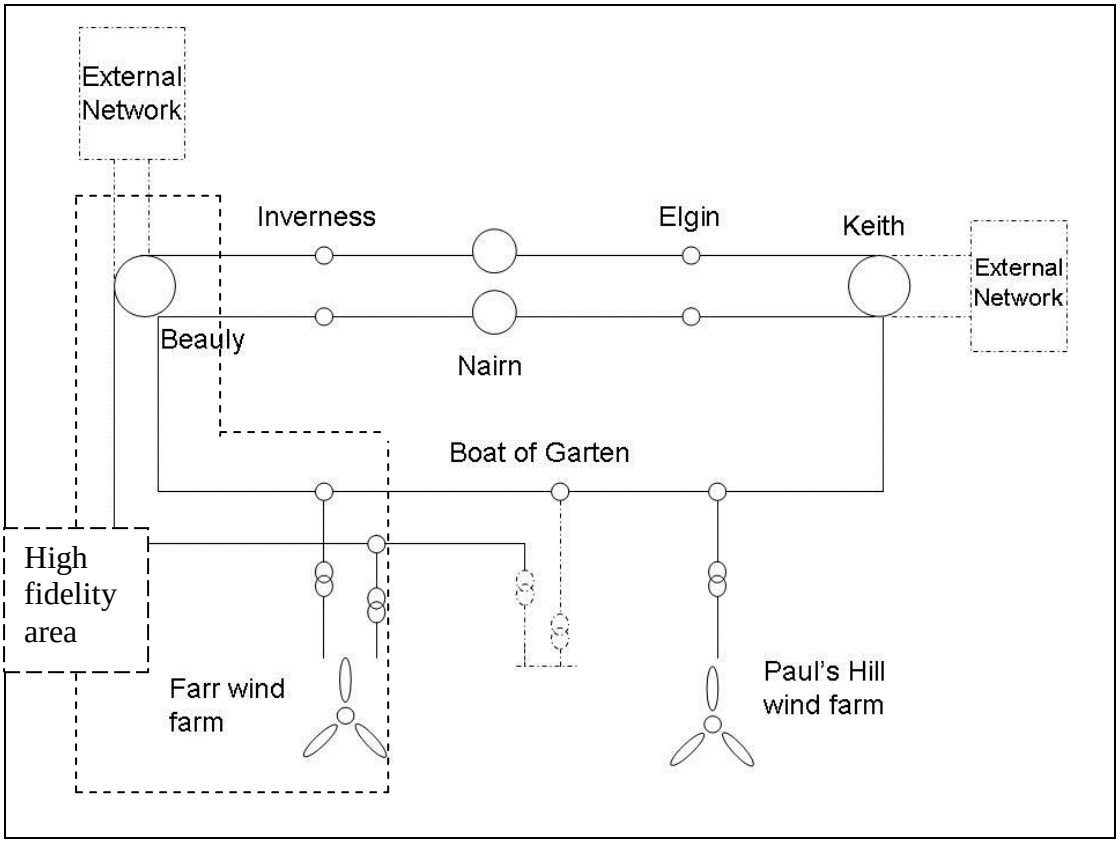


Figure 4.11: The high fidelity area of network modelled

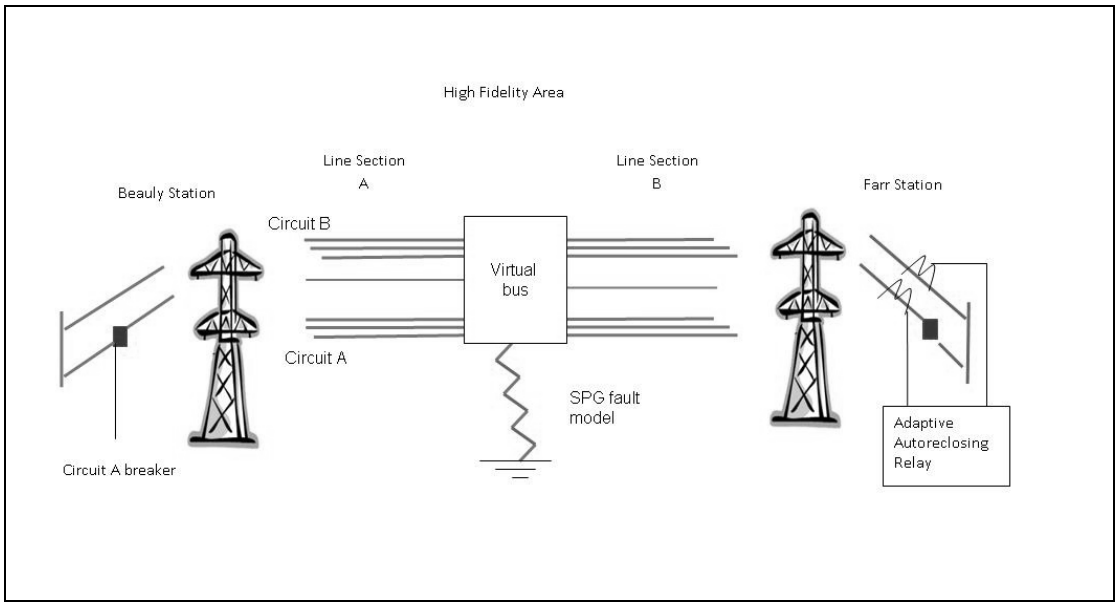


Figure 4.12: The high fidelity section of network modelled

Loads

The loads in the model were assumed to be time independent. For transient studies this assumption is fairly accurate since aggregated loads will not change over a few seconds. Loads were modelled as passive elements, using the shunt resistance and a shunt reactance to absorb real and reactive power. A table (4.2) below is included with the load parameters.

Table 4.2 Load parameters

Bus	Real Power (MW)	Reactive (MVAR)	Equivalent X (Ohms, Ω)	Equivalent shunt R to ground (Ohms, Ω)	Equivalent shunt L to ground (Henries, H)
INVERNESS	77.3	18.4	946.95	225.4	3.01
NAIRN	33.3	10	1742.4	523.24	5.55
ELGIN	54.3	12.5	1393.9	320.89	4.44
BOAT OF GARTEN	21.3	3.7	4709.2	818.0	14.99

Wind farm model

The Farr and Paul's Hill wind farms both use Bonus 2.3 MW machines. The model used in the primary system was developed by RTDS technologies based on the work presented in [48] and is designed to be manufacturer agnostic. The mechanical wind turbine model input wind speed and pitch, and therefore torque, is user adjustable in real time.

The turbine drives a DFIG machine, the basis of operation of which is discussed in the previous chapter. The rotor circuitry and the grid side circuitry is decoupled and governed by two separate vector control schemes. This allows appropriate excitation currents in the rotor, giving maximum energy capture over a wide range of wind speeds and on the grid side, facilitating independent control of active and reactive power. For implementation into the model Scottish system, all the rotor and grid side control circuitry was carefully adapted to reflect the correct system frequency, and other important parameters. Further information on the wind farm control circuitry may be found in appendix 1, section a.

The voltage source converter (VSC) is a back to back converter, consisting of two bridges decoupled by a DC link and shunt capacitor. The firing of the semiconductor

switches is controlled by the small timestep module of the RTDS. The semiconductor devices maybe GTO, IGBTs, MOSFET, thyristors, although their electrical behaviour is essentially the same, switching rapidly from a resistance of around 1000 Ω (open) to 0.001 Ω (closed). A simulation timestep of less than 2μs is necessary to model the control and fast switching of the semi conductor devices. A detailed explanation of how this is achieved is available in [56]. The small time-step module is computationally intensive and thus each VSC must be modelled on a separate GPC processor. There is also a filter bank to eliminate lower order harmonics exported to the grid, and finally a wye-wye connected transformer to step up to transmission voltage. This transformer also serves as the interface between the two time steps, the HV side running at the base simulation rate of 50μs. These components in DFIG machine model may be seen in figure 4.13, a screenshot taken from RSCAD's draft program.

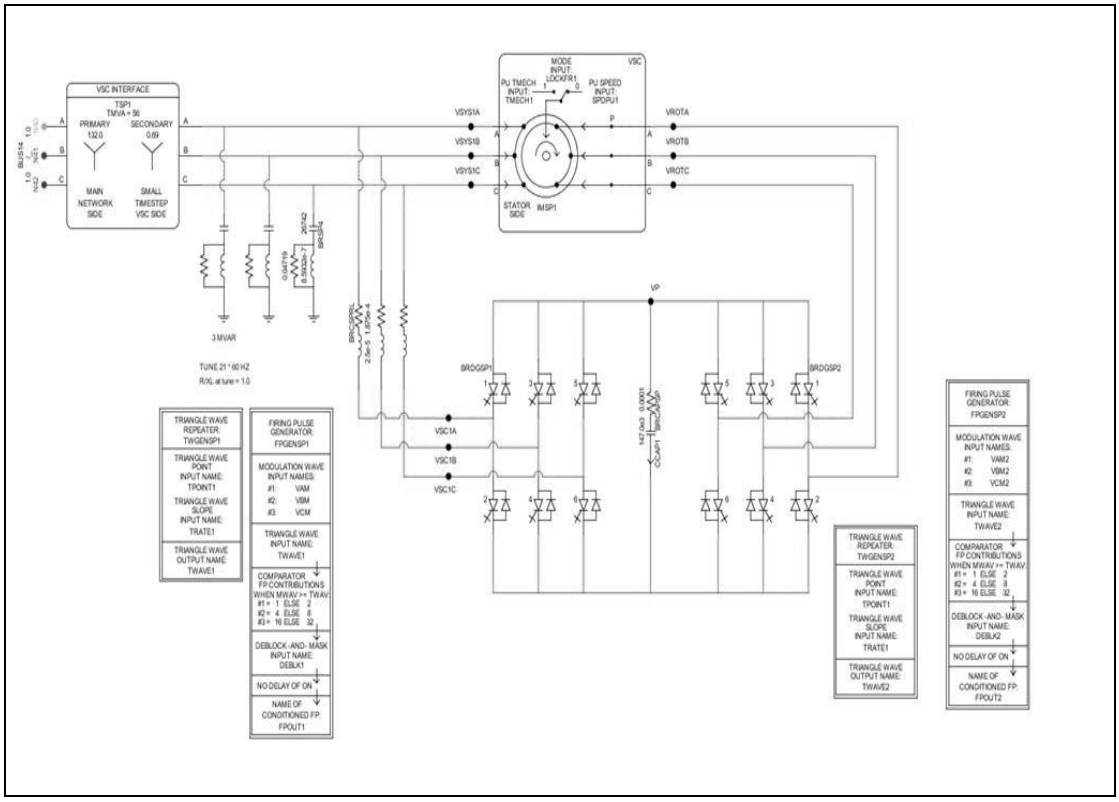


Figure 4.13: The DFIG model in RSCAD draft showing the machine, VSC, filter bank and step up transformer

Model limitations

Wind farm technology is fast evolving and becoming a complex area of mechanical, electrical and control engineering. Moreover, real world designs are manufacturer specific. Detailed transient models are not available since these are proprietary and commercially sensitive. For the purposes of this work, modelling of wind farms in real time was a compromise between processing power, accuracy and commercial neutrality. In order to model a full wind farm with the equivalent number of turbines, the RTDS would need at least 68 GPC processors, in addition to processors for the accompanying cabled collector system and the rest of the primary system. This is clearly not possible with the RTDS resources available for this work. It is thus only feasible to represent a single turbine and scale up to the equivalent size of the entire wind farm. However, the validity of this approach is improved by two considerations. A single back-to-back converter represents the worst case scenario in terms of harmonic pollution. When many converters are represented with a single bridge, all the switches are approximated to operate in step. This is clearly not the case when many bridges exist for several DFIG machines in a wind farm. Although the frequency domain will be similar, the time domain distortion on the waveform should be more onerous when the switches operate in step. Moreover, some wind farms operate a local DC collector network and use a single converter station for AC. From a grid side perspective, this topology is more faithful to the single turbine approximation. For complete fidelity, the cabling system must be taken into account, but it is not possible to represent a short section (e.g. <500 m) of cable with a travelling wave model at the base time step since the propagation time is too fast. Research is currently being carried out at RTDS technologies to make this possible by using a smaller simulation timestep [57]. It should also be mentioned that during grid faults, it is common practice to short the rotor terminals with a crowbar resistance. This protects the rotor side converter circuitry, so that the wind farm can remain connected (so-called ride through capability) acting as an induction generator. This protection methodology was not included in the RTDS DFIG model. However, a pure resistance is frequency independent, and therefore from an AA relay perspective should have a uniform effect across the frequency spectrum.

An important part of the connection to the system is the transformer. In the model, wind farms are represented by a single 132kV/0.69kV step-up transformer, which also

serves as a bridge between the small timestep and common RTDS timestep. Although the topology was not available for the real world wind farms, it may be stated with confidence this representation is unlikely to be realistic. Usually a second step-up transformer would be used and possibly another in parallel to add redundancy to the system. RTDS transformer models neglect resistive losses, modelling transformation through the theory of mutual inductance. The equivalent transformation circuit is highly inductive, and therefore tends to damp high frequencies. Since extra transformation serves as a high frequency choke, this model represents the worst case for adaptive autoreclosing. However, non-linear magnetisation effects may be significant in simulations involving short-circuits such as this. The RTDS manual does not specify whether these are included in the interface transformer model, and if they are, whether non-linearity is accounted for. This is a suggested area for further research.

CVT model

A vital component of any protection scheme is the measurement transducers used to determine the primary system quantities. Representation of the Capacitor Voltage Transformer (CVT) was achieved using the model supplied by the RTDS draft library. Fitton's autoreclosing method [58] modelled the CVT as a low pass filter. An improvement on this approach is to use an equivalent circuit of the CVT, shown in figure 4.14.

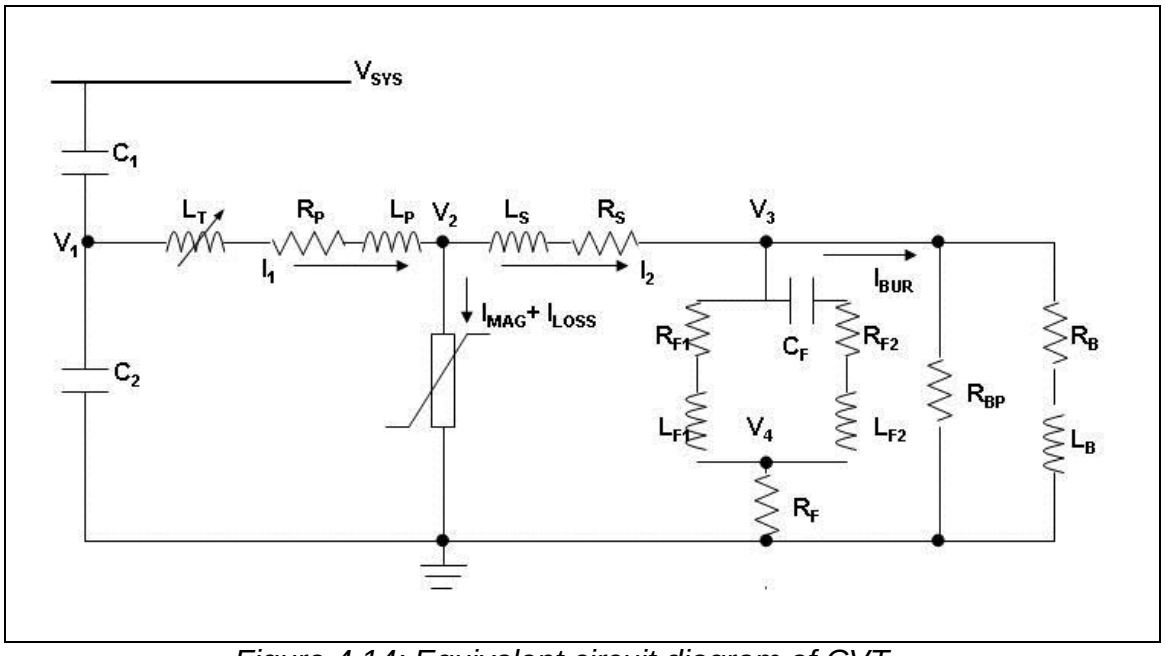


Figure 4.14: Equivalent circuit diagram of CVT

Since the primary system voltages are around a thousand times greater than the desired input to a secondary system device, direct transformation is not practical. A capacitor divider is therefore used to tap off an intermediate voltage, V1, shown in figure 4.14. The adaptive autoreclosing relay is assumed to take an input voltage of 110V.

The RSCAD library includes default values for 230kV transformer. The values of C_1 and C_2 were therefore changed to preserve the value of V1 for a 132kV system.
The system voltage and the primary voltage are related by equation (4.26).

$$V_1 = V_{sys} \frac{C_1}{(C_1 + C_2)} \qquad (4.26)$$

The intermediate voltage is then stepped down with a potential transformer to the desired secondary system voltage V3, after transformation losses are accounted for. It was assumed that saturation and core hysteresis, supplied in the form of B/H curve, would remain the same for a 132kV CVT. A tuning reactance is also introduced to compensate for the phase shift introduced by the capacitor divider. CVT data is notoriously hard to find with the literature scarce, although according to [59], a typical value for CVTs rated in the range 110-500kV is a surprisingly large 42 H. The value of the compensating inductor and the capacitors are related by equation (4.27).

$$L_c = \frac{1}{\omega_0^{\,2}(C_1 + C_2)} \qquad (4.27)$$

Solving the simultaneous equations (4.26) and (4.27) for C_1 and C_2 gives values of 0.0538 μF and 0.1874 μF respectively, denoting the capacitors in the potential divider. The last part of the CVT circuit is filter circuitry to eliminate the ferro resonance caused by the interaction of the capacitor divider and the potential transformer. It was assumed that all other values remained the same given that the desired secondary voltage remained the same as for the RTDS example. However, all inductances and capacitances were scaled accordingly to give the same reactance for a 50Hz (rather than 60Hz) power frequency.

Primary and Secondary Arc Modelling

As mentioned earlier, a realistic arc model is pivotal in the design of any adaptive autoreclosing scheme. In a single phase to ground fault, arcing occurs between the arcing horns on the tower. This flashover is usually initiated by a lightning strike to the conductors, earth wire or tower, and is sustained over the arc path by the system voltage. The arcing behaviour may be separated into primary and secondary stages. The primary arc is in the period before the circuit breakers open and is due to the fault current flowing from the energised phase to ground. The lower current secondary arc is sustained by the mutual coupling between the faulted and healthy phases, and only present when one or more of the phases remain energised. The following is a discussion of salient concepts covered in [32] and [60].

The arc conductance follows the behaviour shown in equation (4.28).

$$\frac{dg}{dt} = \frac{1}{\tau_a}(G - g) \qquad (4.28)$$

Where G is the Stationary arc conductance and g is the time dependent arc conductance and τ_a is the arc time constant. The stationary arc conductance can be physically visualised as the value of the conductance when the external conditions are held constant for sustained periods. It can be evaluated from (4.29).

$$G = \frac{|i|}{U_p l_p} \qquad (4.29)$$

Where i is the arc current, U_p is the average arc–voltage gradient and l_p is the arc length. In the case of the primary arc, the arc length does not evolve with time, and is thus approximately the distance between the arcing horns – taken at 0.5m for the RTDS simulation. U_p is the average arc voltage gradient, and was shown experimentally by Strom [61] to be constant for most of the arc duration for those with a heavy current. With arcs of a peak current between 1.3 kA and 14 kA the value of U_p is constant at around 1500 Vm^{-1}.

Heavy current arc cyclograms of voltage against current also give empirical estimates for U_p, the rate of rise to peak voltage. This is inversely proportional to the time constant. By fitting the cyclograms to equations (4.28) and (4.29) the arc time constant τ_a, can be empirically derived as (4.30).

$$\tau_a = \alpha \frac{I_p}{l_p} \qquad (4.30)$$

If I_p, the peak current is 14 kA, α the value of the proportionality constant at is approximately 2.85 x 10^{-5} m.s.A^{-1}

The secondary arc is an extremely complex phenomena, dependent on a number of parameters. The secondary arc length evolves with time so the previous equations for the primary arc are not valid in this case. Equation (4.28) still holds, but equation (4.29) becomes (4.31).

$$G = \frac{|i|}{U_s l_s(t_r)} \qquad (4.31)$$

Since l_s is now a function of t_r, the time from the initiation of the secondary arc. In the low current range of about 1 – 55A the average arc voltage gradient is actually dependent on the peak current, I_s, and approximately defined by

$$U_s = 75 I_s^{-0.4} \qquad (4.32)$$

I_s is calculated by a steady state estimation of the current, which assumes the arc resistance to be zero. This is a valid assumption since the impedance coupling the faulted and sound phases is in comparison very high, and thus the value of I_s varies little for all values of arc resistance.

Going through the same process with experimental cyclograms, but in the lower current range, the secondary arc time constant is derived as (4.33).

$$\tau_s = \beta \frac{I_s^{0.4}}{l_s(t_r)} \quad (4.33)$$

With β = 2.51 x 10^{-3} m.s.A^{-1}. As the alternating voltage, induced by the healthy phases, passes through successive zeros, the secondary arc extinguishes and re-strikes a number of times. This is the most difficult characteristic to predict. In the secondary arc model developed in [32] a number of different systems were simulated to empirically derive the following equation for the arc re-striking voltage (4.34).

$$V_r(t_r) = [5 + \frac{1620 T_e}{(2.15 + I_s)}] (t_r - T_e)\, h(t_r - T_e) \qquad (4.34)$$

Where T_e is the time from the initiation of the secondary arc to a current zero and $h(t_r - T_e)$ is the delayed unit step function. Following the arc voltage reversal (i.e. a voltage zero), the current is held at zero until the voltage exceeds the level specified in equation 8, and the arc begins to conduct again. Final extinction of the arc occurs when the voltage fails to exceed this re-striking value $V_r(t_r)$.

The duration of the secondary arc depends highly on the wind speed, since this clears the ionised arc products and inhibits further arcing over the arc path. At relativity low wind speeds, 0-1m/s, arc elongation follows the time dependence (4.35).

$$\begin{aligned} l_s(t_r) &= l_s(0) && \text{for } t_r \le 0.1 \\ l_s(t_r) &= 10 t_r\, l_s(0) && \text{for } t_r \le 0.1 \end{aligned} \qquad (4.35)$$

Where $l_s(0)$ is the length of the secondary arc at time 0. It should be mentioned that whilst wind speed is not taken into account, low wind speeds are assumed in this work since they lead to the longest secondary arc, and thus the worst case from a system restoration perspective.

RTDS Arc implementation

RTDS supply a precompiled fault arc model component that models both the primary and secondary arc. A detailed explanation on how this model is realised is described in [62]. Briefly, the arc conductance is solved in each timestep and this determines the value for a time varying resistance. Integrator blocks are used to solve differential equation (4.28), and each of the parameters that it depends on, in separate primary and secondary stages. In the case of the secondary arc where the arc length evolves quickly with time, the arc length is simulated using random number generators. The work in [62] shows the arc model to have good agreement with staged fault tests and those carried out in ATP draw, and thus is acceptable for the purposes of this work.

Permanent fault resistance in RTDS

In the case of a permanent fault, the fault path from conductor to ground was assumed to be a default value of 2 Ω and purely resistive. The permanent fault resistance was varied from 0-100 Ω for training the neural networks, and this is described in more detail in chapter 6.

Fault and circuit breaker control

The methodology for originally tripping the circuit breaker (i.e. protection algorithm) is not in the scope of this thesis. Assuming ideal operation of the first main protection, and correct selection of the faulted phase, the fault signal and the subsequent initial circuit breaker trip can be derived from the same signal in runtime. Following a line fault, the circuit breakers are assumed to open simultaneously at the Farr and Beauly substations, their operation derived from the same signal.

Figure 4.15 shows the breaker control system in RSCAD draft. With reference to figure 4.15, the faulted phase voltage is imported to the subsystem and fed to a zero crossing detector, which emits a binary signal on a negative to positive zero crossing. When the 'transfault' button is pressed, a binary signal is sent to the AND gate, sending a binary when both the button is depressed *and* a zero crossing is detected. The flip/flop module then is activated emitting a permanent 1 until the reset button is pressed. The signal is then subject to a delay depending on where the fault inception point on the waveform is desired.

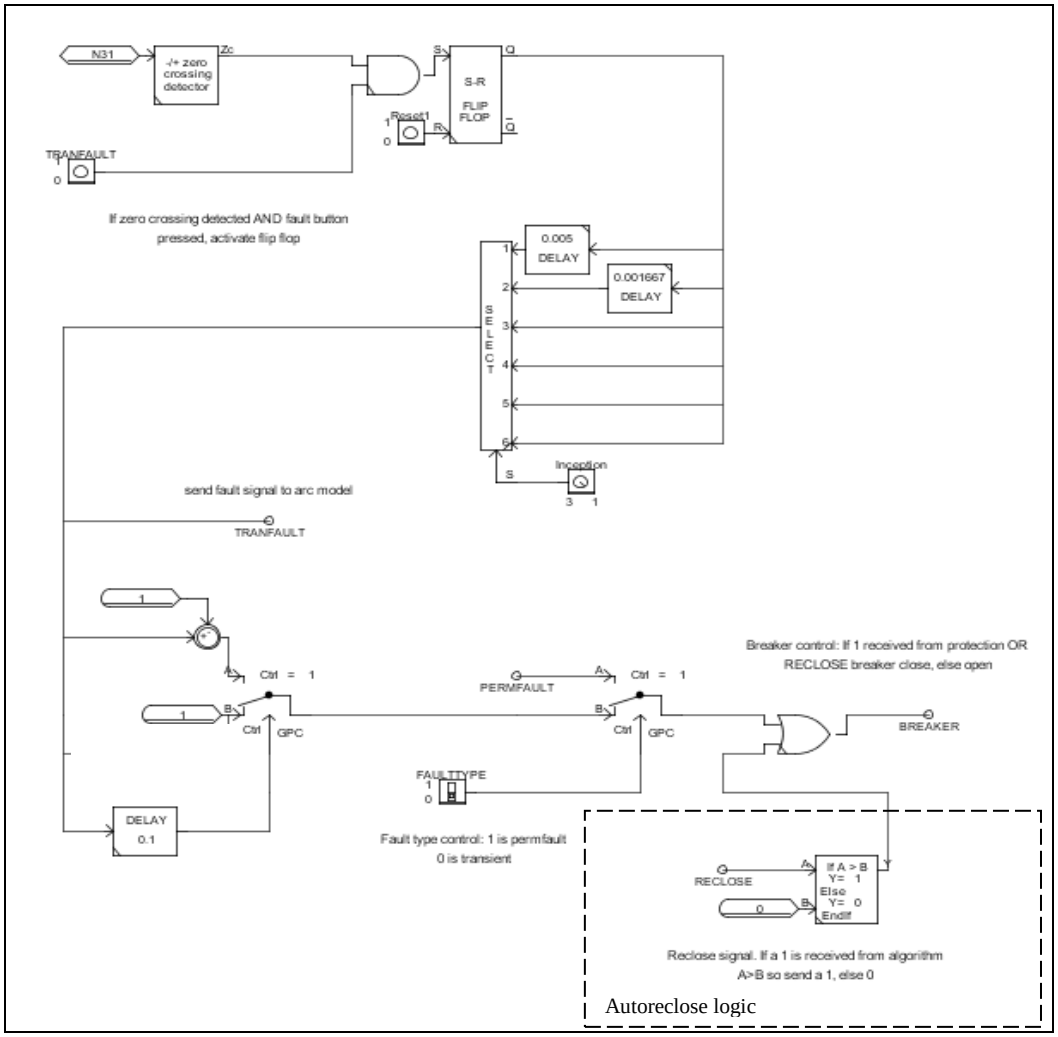


Figure 4.15: Fault and breaker control

The inception point is selectable in runtime using a dial. A quarter of a cycle (0.005s) corresponds to a maxima and a half maximum (0.001667s). If a minima inception point is desired, the signal is not delayed. The signal is then fed to a fault arc model (or permenant resistance in the case of a permanent fault). The same signal is then delayed by 5 power system cycles, (a conservative response time for modern circuit breakers at 132kV) and converted to logical false so that the breaker will be set to the open state. An identical control subsystem that first sends a signal to a permanent fault (and bypasses the transient fault logic) may be selected at this point. This switch is selectable in run time. If the breaker control signal is 1 it is closed, else it is a 0 in the open state. The breaker closed signal may be from the fault breaker logic or (via the OR gate), the autoreclose signal, received from the autoreclosing algorithm. Further details of the latter may be found in chapters 6 and 7.

4-5 Conclusion

This chapter outlines the theory of real time digital modelling, discusses the RTDS hardware and the 132kV model of part of the Scottish system. Although the 132kV system is constructed to the highest possible fidelity with the available hardware, system model limitations are also dealt with. A complete view of the primary system model is available in appendix 1a.

Chapter 5 – Artificial Intelligence in power systems

5-1 Introduction

Having established a primary system model, the challenge is then to build a suitable secondary system that delivers the desired functionality. In the real world, the parameters that determine the faulted transmission line response are subject to uncertainty:

- What causes the fault?
- What is the fault path resistance?
- Is the conductor shorted through a tree or is it broken and touching the ground?
- In the case of lightning strike: where was the strike?
- How near was it to the conductors?
- What are the atmospheric conditions and wind speed that determine the arc ignition characteristics?
- Is the high frequency content due to arcing or is the transmission line acting as a giant antenna?

Moverover, the complex interplay of these parameters has non-linear effects on both the arc signature, and the permanent fault signature. Although every attempt is made to keep the simulation as accurate as possible, there will be limitations in comparison to the real world.

An adaptive autoreclosing relay is a prime example of smart grid technology, in that *it carries out intelligent automated changes to the power system in real time, to suit conditions as they evolve*. In order to achieve this functionality in protection, it is necessary to apply human judgement in a few power system cycles. This is a non-trivial task. Over the last two decades, researchers in computer science and engineering have begun seriously considering how to solve real world problems such as these by studying the principles of artificial intelligence (AI). It is therefore necessary to digress from the main thrust of this thesis in order to cover AI methods

and discuss how they may be applied to the autoreclosing problem. AI methods are sometimes referred to as ‘soft computing’, as this term does not evoke associations with the realm of science fiction.

5-2 Artificial intelligence in power systems

Since the interconnection of power systems began at the end of the nineteenth century, society has enjoyed increasingly reliable and economic access to electrical power. This evolution of power systems has seen transmission and distribution emerge as separate fields from that of generation. Over this period, many challenges have been solved, but also new ones have emerged. Presently, increasing concerns over climate change are acting as a catalyst for change in power systems, and demanding a concomitant rise in complexity. Some of the present issues in power systems are voltage collapse, system stability, load forecasting, inter-area oscillation, power quality, effective market design, charging methodologies, distributed generation and the coordination of protection and control [63]. The more complex the primary system, the more difficult it is to tackle these challenges with simple logic and algebra. Many emerging problems are difficult to frame such that a procedural, algorithmic computer program may be used to resolve them. AI is an area of computing that is suited to such problems.

The main examples of artificial intelligence applied to power systems thus far are, evolutionary computing (most notably Genetic Algorithms), fuzzy logic, artificial neural networks, expert systems and multiagent systems [63]. Each area is more suited to some problems than others, but there are common themes among the techniques. They all take their conceptual inspiration from nature, they are robust in the presence of error, noise, complexity and non-linearity, and they are suited to dealing with problems that usually require human reasoning. AI techniques often necessarily involve abstract concepts, and there is some vagueness, crossover and redundancy between them. The forgoing discussion cannot therefore be exhaustive or definitive.

A cautionary note should be made when considering AI and the safety criticality of power systems, especially in the realms of protection, operation and control. When

applied in these areas, AI systems must guarantee correct operation of the equipment in every possible scenario. In recent years, there has been a tendency in the literature to attempt to apply in vogue AI techniques in cases where they are clearly not suited to the application. It would not be fair or in this author's interest to pick on any specific examples at this point. However, figure 5.1 gives a crude indication of IEEE publishing trends in AI techniques, by showing the chronology of papers published containing these terms in the title or key words. After an initial boom period, publishing activity in these areas is declining, apart from genetic algorithms which has slowed but remains stable.

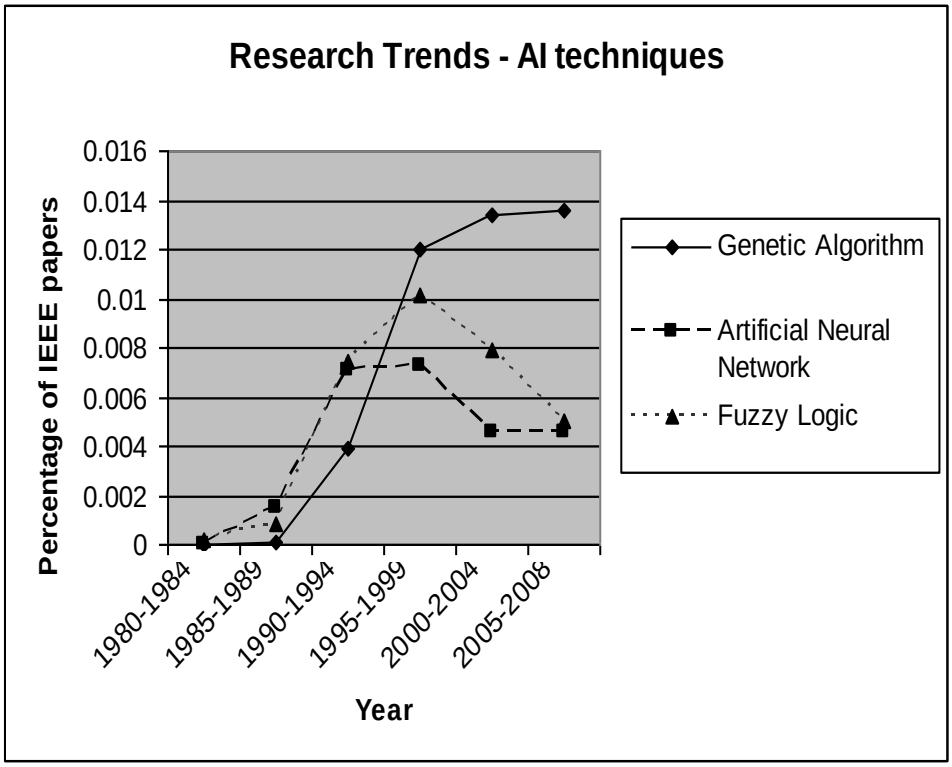


Figure 5.1: Research trends in AI: percentage of IEEE papers containing these AI techniques in the title or key word section

5-3 Evolutionary Computing and genetic algorithms

Evolutionary computing (EC) is a branch of computer science loosely inspired by evolution in nature. EC algorithms usually employ some sort of weighted random selection process to optimise an objective function, a process known as metaheurism. Each proposed solution requires a quantitative assessment for its ability to solve the

problem, which is the purpose of the objective function. The stochastic nature of metaheurism makes it useful for complex search spaces where many local maxima exist, although they do not guarantee an optimal solution has been found. Power system problems often involve complex, multidimensional and non-linear search spaces that cannot be explored by straightforward calculus or hill climbing. Over 95% of papers published in the area of evolutionary computing involve genetic algorithms [64], although other applications exist, notably simulated annealing, particle swarm optimisation and classifier systems. Simulated annealing (SA) is a global optimisation technique suited to a search space made up of discrete data. In each step of the algorithm, a solution is randomly replaced with a nearby solution in the search space. The probability of selection is dependent both on the new value's difference to the current solution and on a global probability, 'temperature', the latter being gradually reduced throughout the process. A high temperature is set to allow uphill moves in the search space, as well as downhill, so the algorithm does not stick on local minima. As temperature is reduced, selection becomes less random and tends to converge on the global minima. Particle swarm optimisation (PSO) is a relatively new EC technique where a fixed population of particles roam around a search space, each with a given velocity. The velocity iteratively gives a particle's new position within the search space, and is in turn determined by the particles own best known position and the global best known position. The significance of each term in the velocity calculation is weighted by the user, and has an important bearing on the success of the algorithm. These weighting parameters may also be optimised by another technique known as meta-optimisation.

Genetic algorithms (GA) are a global parallel search and optimisation technique based on Darwinian evolution in nature. They are a versatile and elegant technique, with wide scope for application, which may explain their continuing popularity following the AI research boom of the 90s (see figure 5.1). GAs normally involve the following basic steps:

- Initially, a population of individuals is chosen at random, each representing a possible solution to a problem.
- The elements to the solution are encoded, usually as a binary string, so that each individual solution is represented as a chromosome or genome by its

constituent “genes”. For example, if 01 represents blue and 11 represents red, then an individual, “bluered” may be defined as 0111.

- Each individual is then assessed for fitness against a fitness (objective) function, which measures the quality of the proposed solution.
- A new population of individuals is formed by ‘selectively breeding’ the old population. This breeding process can take many forms, but usually two individuals are selected by a weighted random process. This is also known as weighted roulette wheel selection, where segments of a hypothetical roulette wheel are assigned different sizes.
- This new population is assessed for fitness and if the convergence criteria for any single individual is not met, the process begins again, albeit with a population of ‘fitter individuals’, or better solutions.

A classifier system or learning classifier system (LCS) is a machine learning tool that is able to dynamically categorise its environment in order to apply the most appropriate rule for a desired behaviour. An LCS is useful when behaviour can be measured for its effectiveness, but it is not initially obvious what constitutes good behaviour before empirical testing. This is the case where environments are complex and/or evolving. An LCS is commonly implemented as an extension of a genetic algorithm, in that a series of IF-THEN rules is also encoded into the bit string of the proposed solution, and measured for fitness. Using the earlier example, a ‘bluered’ 0111 solution may be augmented such that this chromosome leads to execute or ignore a series of actions denoted as ones or zeros. For example, 011100 could be decoded as “don’t do A *or* B” where as 011101 may indicate “do action A *not* action B”. Such an example is trivial, but with a more complex problem high fitness rules may emerge that are not obvious from the outset.

Application to the autoreclosing problem

How might genetic algorithms or other EC techniques be applied to the adaptive autoreclosing problem? In terms of the classification of the secondary or primary arc, there is little scope. This is because it is difficult to frame these problems as optimisation. Given that a transient fault has been diagnosed, when is the optimum time to reclose? As mentioned in chapter 2, there is a further consideration of the

effect of autoreclosure on transient stability. For a single synchronous machine feeding an infinite busbar through a double circuit line, it has been shown [28] that autoreclosing should take place when the swinging postfault angle difference is at the prefault angle difference. However, in multi machine systems the situation is more complicated. In some cases, [65] shorter reclosure times caused the critical clearing time to suffer. Indeed, stable autoreclosing in such a complex multi-machine system may be better tackled with AI techniques over deterministic numerical methods. To what extent is the situation altered for systems with high penetration of variable speed wind turbines? Both DFIG and FC machines are to some extent decoupled from the power system so this further complicates the problem. EC techniques could potentially be helpful in solving optimal autoreclosure times for this more complex topology. However, it is unlikely that they could be deployed in a truly online time-critical protection relay because they are relatively computationally intensive even for modern processors. This is partly because EC techniques are quite abstract so the stages are coded with high-level routines, which then must be iterated many times to create populations of solutions. Moreover, the inherent randomness of EC techniques gives no guarantee of when the stopping criteria are likely to be met. It is possible they could be used in a 'pseudo-adaptive' capacity by optimising a number of possible sequences offline, for a given set of system conditions. GAs could also be used to optimise the primary technique, but such hybridised approaches to AI will be discussed later in this chapter.

5-4 Multiagent systems

In multiagent systems (MAS), each agent is an autonomous computer program that exists in a complex dynamic environment, and interacts with other agents to achieve the functionality desired. Each agent does not require knowledge of the entire system, and they can therefore be useful for problems that monolithic, hierarchical systems find challenging. In addition, MAS may exist in a physically distributed nature on a number of devices [66].

Application in power systems

In power system protection terms, a single agent could therefore exist on several physical devices. IEC61850 GOOSE messaging between IEDs on the substation bus

also supports this approach [67]. This is because objects may be created in the software domain, and actually exist in a dispersed nature: over several IEDs connected with a fast ethernet ring bus. As mentioned in chapter 2, [31] describes a multiagent system applied to adaptive autoreclosing where the agents calculate optimal reclosing time, transient energy and arc extinction. Debatably, this functionality could be achieved by simply running three parallel algorithms and using logic gates to issue a reclosure condition. True MAS are useful in a decentralised architecture, in that no one element has full control of the system because it is too complex to understand. Applying MAS to a specific relay element, such as adaptive autoreclosing, is hard to envisage, unless an agent is required to externally collect remote data from other parts of the substation, or wider power system. From this perspective, MAS are more suited to holistic and abstract wide area applications such as load shedding and islanding schemes. Even so, their behaviour must be guaranteed within acceptable limits, which suggests the need for a supervisory element, and this strictly speaking, is defined as a monolithic system. One promising MAS application maybe in a substation supervisory layer or 'adaptive protection controller', a layer to supervise and control various adaptive elements of IEDs within a substaion. From the autoreclosing perspective, agents could be responsible for farming remote data: from the control room, or the weather via internet, and blocking autoreclosure should it not be appropriate.

5-5 Expert systems and fuzzy logic

Expert systems and fuzzy logic are heavily interrelated. Expert systems (ES) attempt to address specific problems that would usually require a human expert by using a rule base, a working memory and an inference engine. Expert systems allow human reasoning to be applied extremely quickly, and the knowledge of several experts can be pooled for a single system. The rules may be changed or updated without having to rewrite the initial rule base. The advantages over human experts are increased availability, speed, cost, reliability and reproducibility. From a protection perspective, speed and availability are the most exciting prospects. For example, a complex switching operation following a fault condition can be implemented on millisecond timescales, rather than requiring human intervention from the control room. An expert system is debatably an overarching term for any form of decision making AI, but for

this thesis, an ES is defined specifically as a rule based ES. A common criticism of the rule based ES is that it relies on an expert's ability and willingness to distil his or her judgment and behaviour into a series of IF-THEN-ELSE rules.

ES failures may be attributed to sabotage (in that the expert is concerned that such a system would make them obsolete [63]) but are usually because human reasoning concerning complex problems is an imprecise and intangible process. Fuzzy logic attempts to address this problem by allowing for the vagueness and uncertainty in the reasoning process. In Boolean logic, values can only belong to one of two binary discrete states: yes or no, on or off, 1 or 0 and so on. For example, the answer to the question "Is the weather good today?" may only be addressed by Boolean logic as yes or no. This may not be a satisfactory way of describing the day, especially in the UK, where the weather systems are subject to rapid variation. On a given day it may be sunny in the morning, cloudy in the afternoon and raining in the evening. Fuzzy logic attempts to address the shortcomings of 'crisp' conventional logic by assigning a degree of truth to statements about variables. The way in which a fuzzy variable's degree of membership varies with a given parameter is described by its membership function [68]. An example of a fuzzy membership function is shown in figure 5.2.

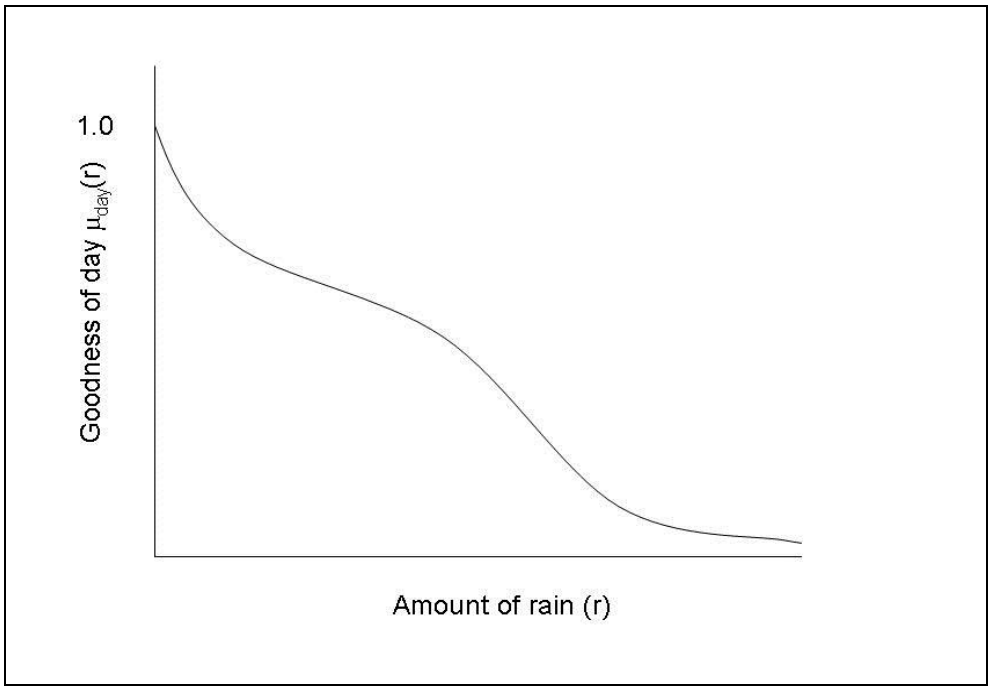


Figure 5.2: A fuzzy membership function: the day's quality depends on rain

Simply put, the day may be described by fuzzy logic as 33% good, after say, 5mm of rain. A human might sum the day up linguistically by describing the day as 'fairly poor'. Clearly, the relation between '33% good' and 'fairly poor' is more accurate than the nearest crisp classification 'the day is not good today'. But this approach does not preclude later augmenting the model to include the fuzzy variable's dependency on some other parameter. For example, in a human's assessment, 'goodness of day' may also depend on hours of sunshine.

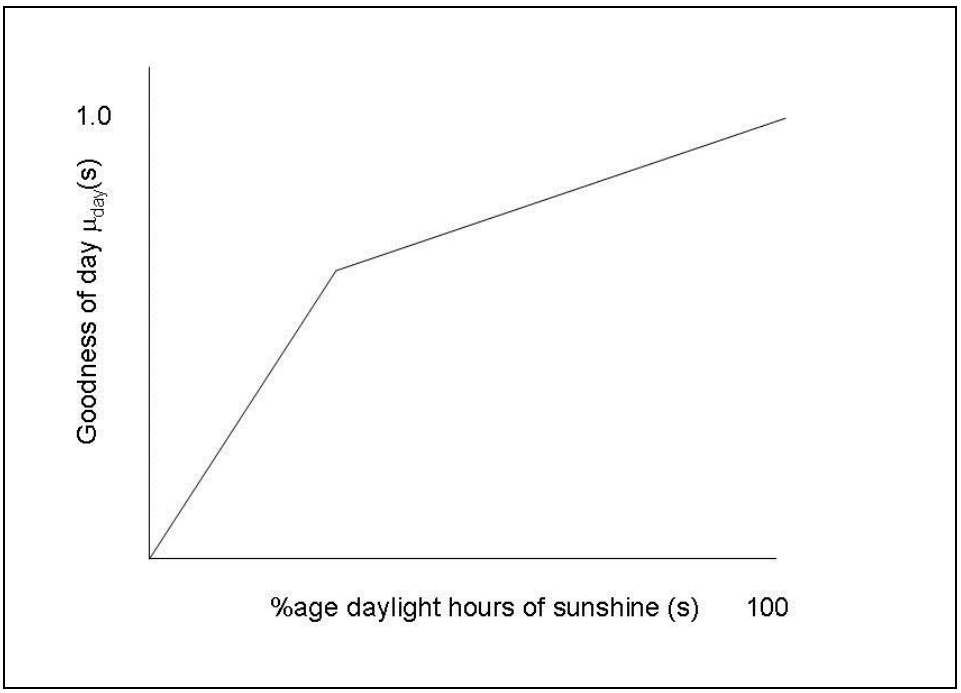


Figure 5.3: A fuzzy membership function, the day's quality depends on sunshine

What happens if following a sunny morning, i.e. 40% daylight hours of sunshine, an unusually large amount of rain, >5mm, falls in the afternoon? Using crisp logic at this point presents a quandary about whether the day is categorised as good or bad. Due to both dependencies, the day may be described both as good and bad, and neither is correct. However, in a fuzzy system, taking into account how the quality of the day varies with these two parameters offers a more holistic assessment of the situation. The percentage quality of the day (the day's 'goodness') may be mathematically described as

$$Q_{day}(s,r) = 100 \times (\Phi \mu_{day}(s) + \theta \mu_{day}(r)) \qquad (5.1)$$

Where Φ and θ are weighting coefficients between 0 and 1 describing the importance of each dependency. Each membership function is normalised between 1 and 0 but since the percentage 'goodness of the day' $Q_{day}(s,r)$ must lie between 100 and 0,

$$\Phi + \theta = 1 \qquad (5.2)$$

In a fuzzy system, the inputs must be first 'fuzzified', and the outputs 'defuzzified' to extract meaningful information. In the process of fuzzication, an input is designated a degree of membership to each fuzzy variable in a way that supports the resolution of the problem in hand. In the example, 5mm of rain is 'fuzzified' such that it has a 33% membership to the fuzzy variable 'goodness of day'. In a fuzzy system, this input might also determine another fuzzy variable such as 'need for umbrella'. Defuzzification is the way in which meaningful output is derived from the fuzzy system, which is highly dependent on the task in hand. Usually, fuzzy variables are combined to arrive at a decision that can only be described with crisp logic. Adhering to the ongoing example, the fuzzy system could be defuzzified to decide if the user should drive to work.

A useful property of fuzzy logic is its ability to quantify linguistic variables, known as hedges, in the inference engine of an ES. Hedges are typically adverbs such as *very*, *slightly* and *quite*. As is shown in the example, this is a much more accurate way of describing knowledge about something than crisp logic. However, as with a basic ES, a human expert must be available, they must understand and be able to articulate the thinking process, and a competent software architect must be available to program the same.

Application in power systems

ES and fuzzy logic are useful when trying to mathematically represent a human decision making process when faced with a problem. There are many examples of ES applications in power system problems, which are discussed further in [69]. Once designed, they are fast to execute and can thus be realistically deployed online. In

power system protection, specifically *fuzzy* ES applications, they are usually hybridised with some other feature selection technique. These range from fault classification and location in transmission lines, [70], [71] and [72], distribution networks [73] to transformer protection [74]. The difficulty in the adaptive autoreclosing problem is that the arcing phenomena is not one that is readily describable with language. The task of detecting an arcing fault is essentially one of pattern recognition, and given that an arcing fault exists, one must detect the extinction of the secondary arc. With reference to chapter 2, figure 2.1, the presence of the secondary arc is classified by high frequency content on the faulted phase. However, this is also initially present in the case of a permanent fault, figure 2.2, due to the circuit breaker operation and other harmonic pollutants. Although a human with the requisite technical knowledge can easily tell the difference on inspection, it is difficult to describe, and then *quantify,* a fuzzy ES for distinguishing between these two scenarios.

A separate consideration in reclosing time is one of optimal stability. In a multi-machine system, varying reclosing time will affect the stability of each generator relative to the others, and by differing amounts. The situation is further complicated by systems with high penetration of wind power [75]. This trade-off for optimal stability could be managed by a fuzzy system.

An attractive property of fuzzy expert systems is ease of augmentation. A system may undergo a change in topology, or over time, operating experience or statistical data may accrue giving a better knowledge base. In relay terms, rules may be added by substation remote link, without the need for a complete reinstallation. Lastly, fuzzy logic may address considerations based on external factors, such as the weather or altitude of the transmission route at the location of the fault. For example, thunderstorms will increase the chance of lightning strikes and thus temporary faults, and higher transmission towers are more exposed. Winds will decrease secondary arcing times but very high winds will also increase the chance of permanent faults due to downed lines.

5-6 Artificial Neural Networks

Artificial neural networks (ANNs) are inspired by the workings of the brain. A number of simple processing units are wired together in an interconnected network to perform a more complex task. The architecture of the network is dependent on the problem in question. ANNs are analogous to the brain in several ways.

- They are tolerant of failure in that a single neuron should not affect the network's performance.
- They require training in order to evoke the desired behaviour
- They are able to generalise, recognising trends when presented with similar data not used in the training.
- They are highly parallel (although in practice this is rarely exploited since the ANN is programmed in software and individual calculations are carried out by a CPU in serial steps).
- They excel at discerning non linearity, are tolerant to noise, fuzziness and uncertainty.

However, there are also some important differences

- Brains function with complex bio-electrochemical processes involving many types of neurotransmitters and receptors, whereas ANNs are purely electronic, existing in software or hardware.
- ANNs are considerably smaller and simpler than their biological counterparts. For example, the human brain contains around 10^{11} nerve cells of varying types and architectures. ANN sizes vary hugely but most applications use no more than 10^{3} neurons [76].

<u>The ANN Neuron model</u>

In nature, nervous systems, including the brain, consist of many nerve cells wired together by synaptic links. A neuron has a fibre-like connection called an axon. The axon commonly splits into many links that terminate in synapses, each connected to other neurons. Electrical pulses in the axon determine what chemical

neurotransmitters are released by the synapse. In a complex process, these act on receptor cells, which in turn determine the potential across the next neuron membrane. When the membrane reaches a certain potential (the action potential), an electrical pulse is transmitted to further neurons downstream via their axons. When the action potential is reached, the neuron is said to be "fired".

The fundamental component of an artificial neural network is a neuron. Each single neuron consists of inputs, the computational centre and its outputs. Depending on the architecture of the network, a neuron may have many or few inputs. Each input has an associated weighting coefficient, w_{kp}. This variable establishes the relative importance of each input. The output also has bias, θ_k. The output of a neuron with n inputs is usually defined as equation (5.3).

$$y_k = f\left(\left(\sum_{p=1}^{n} w_{kp} x_p\right) + \theta_k\right) \qquad (5.3)$$

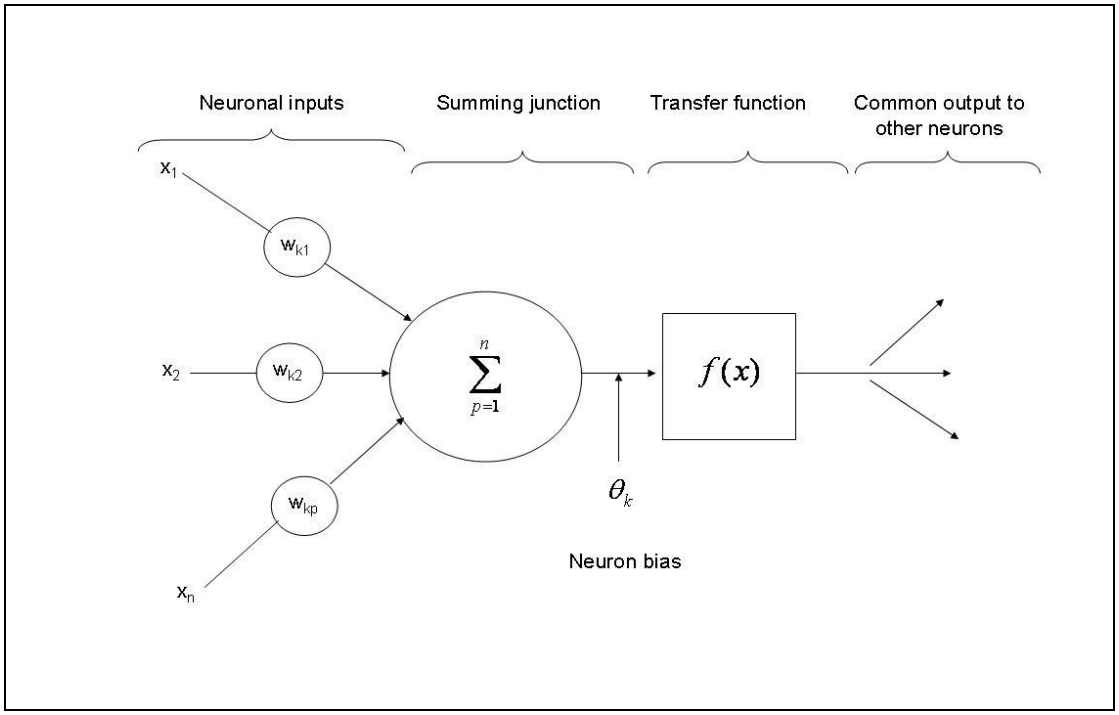


Figure 5.4: The neuron model

Where f(x) in (5.3) is the transfer function. This maybe a simple threshold, such that summed input values over a certain threshold lead to output, i.e. the neuron 'firing'.

More often, these are some sort of continuous function. It may simply be a pure linear transfer function. Other common functions are logistic sigmoid, equation (5.4).

$$f(x) = \frac{1}{1+e^{-x}} \qquad (5.4)$$

Or the hyperbolic tangent, equation (5.5).

$$f(x) = \tanh(x) \equiv \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \qquad (5.5)$$

For any real value of x, f(x) is between 0 and 1 in a logistic sigmoid and -1 and 1 in the case of a hyperbolic tangent. In effect, the transfer functions normalise the input values to any downstream neuron. (5.4) and (5.5) are plotted with a linear function in figure 5.5.

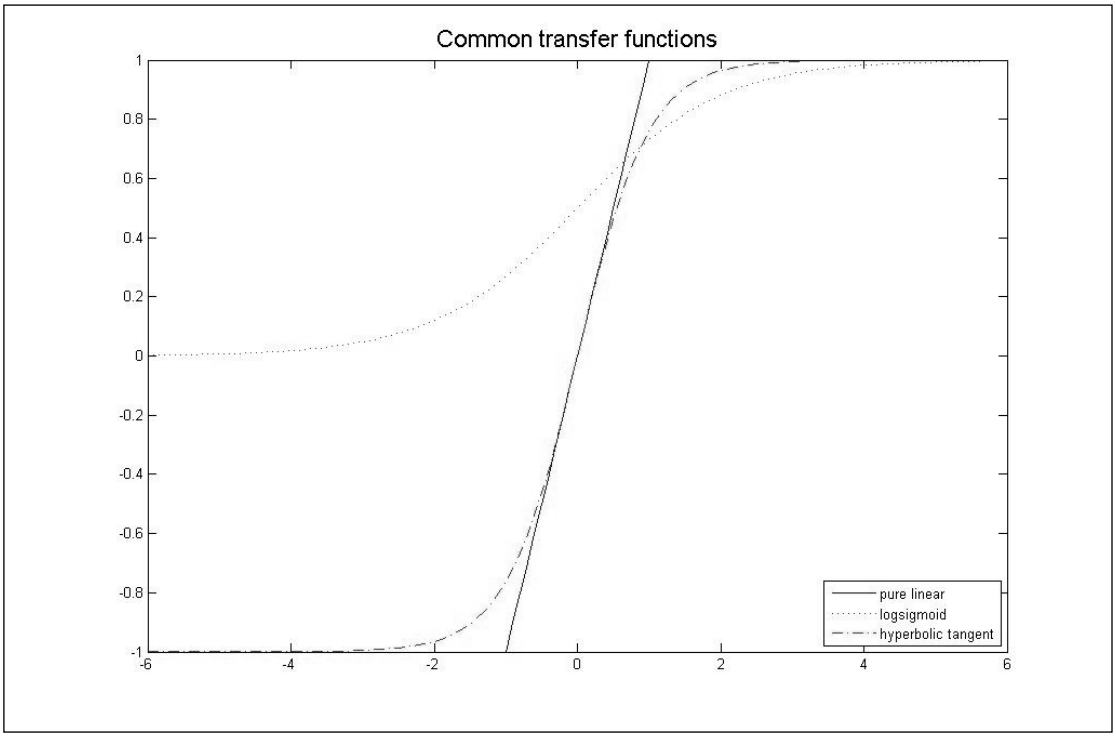


Figure 5.5: Common transfer functions

For determining non-linearity between input and output data, the transfer function must be non-linear. The output of each neuron may be relayed to one or several neurons downstream in the network.

Network training

Networks must be trained to perform in the desired way to solve the problem in hand. This usually involves taking input data for which the desired ANN response is known, and adjusting the values of each synaptic weight, w_{kp}, so it performs in the way that is required. Several algorithms exist for this, such as backpropagation and gradient descent. A training set is divided into a subset of training cases, A, and a smaller subset of test cases, B. The network is trained using input set A, and the desired outputs, and then tested with set B. If the error produced by the ANN when subject to B is similar to that produced by A, the ANN is able to generalise - in other words, it has 'learnt' rather than memorised. This is so because it is behaving in the desired manner for data it has not seen before. This type of training is "supervised" in that the network is shown the correct behaviour by being presented with correct outputs for a given set of inputs. The example correct outputs used in training are often called target vectors. In 'unsupervised learning', the ANN is not given any target vectors and the training algorithm serves to reinforce clustering in the input data. This is useful when the correct behaviour for the problem is not available.

Neural Network Architectures

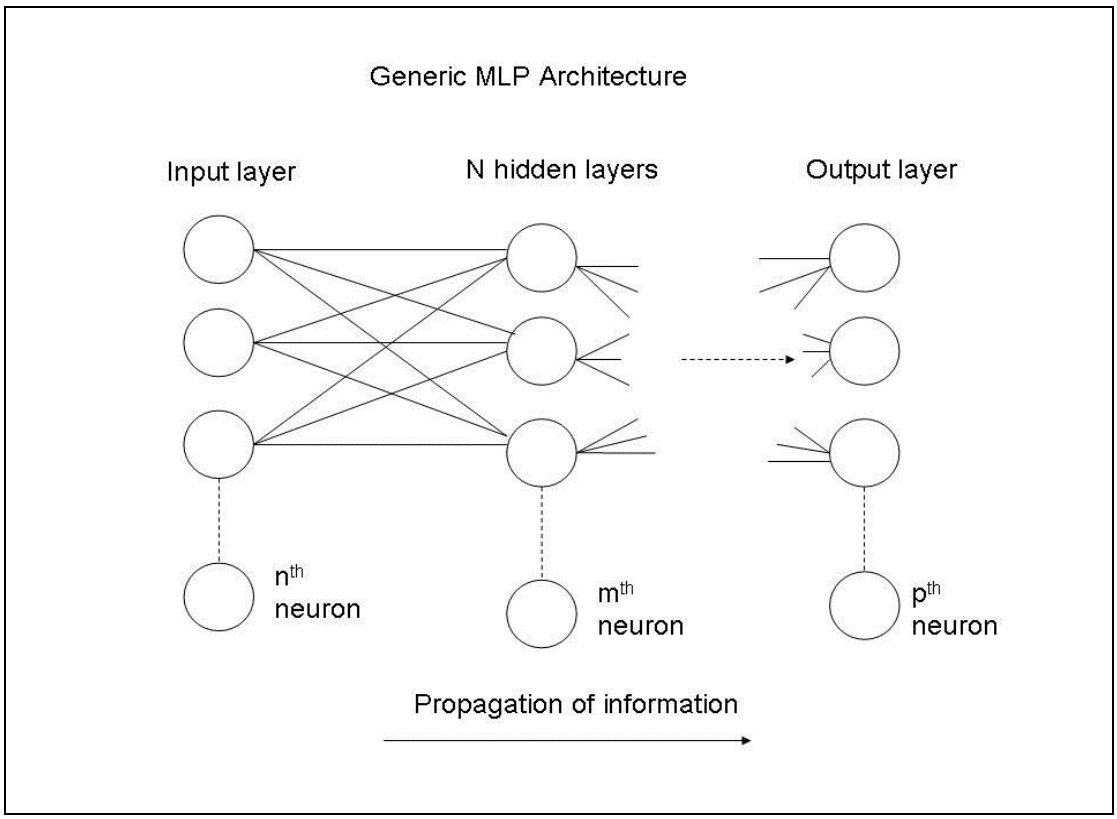


Figure 5.6: MLP architecture

Many network architectures have been developed, each suited more or less to a wide range of problem types. Figure 5.6 shows a multilayer perceptron, (MLP) [77].

Pattern recognition architectures

The simplest architecture is the perceptron. The neurons are arranged in two layers, inputs and outputs, and each neuron is fully connected to every neuron in the next layer. The perceptron is trained to divide the inputs into two classes. Although simple, the ANN is only able to determine relationships in input classes that are linearly separable. If a straight line on a plane (or plane in a space …etc*) cannot be drawn between the classes in the input vectors, then a perceptron will not be able to successfully classify all of the inputs. When non-linearity exists in the class boundaries, one or more 'hidden layer' must be introduced, and the network is defined as a multi-layer perceptron (MLP), shown in figure 5.6. The transfer functions in each layer of an MLP must be non-linear or the hidden layer becomes redundant, as it is possible to represent them with an equivalent single layer perceptron. Perceptrons are feed forward networks, in that all information flows in one direction and there are no feedback loops. They must be trained with supervised learning techniques, where a desired behaviour is shown for given inputs and the weights adjusted accordingly by a suitable algorithm.

Hopfield network

A Hopfield network is a recurrent network that is able to perform the task of associative memory [78]. The network architecture is recurrent in that output is iteratively fed back into the input. The network tends to converge to a local minima state. In the training process, these local minima are arranged to be the states that the network is required to remember. The network then becomes a content addressable memory system, in that it converges to a state with which it most associates its current input. The inputs do not have to be exact and thus the networks are tolerant to noise and incomplete data. Periodic or chaotic behaviour is avoided by keeping the weights symmetrical.

* The analogy continues for N dimensional space. For linear separation in N dimensions, there must exist a linear hyperplane described by N-1 dimensions than can separate the classes

Kohonen network

A Kohonen network [79], also known as a self organising map (SOM), is a way of visualising classes in higher dimensionality input data as a two dimensional map. Kohonen networks use unsupervised learning to find clustering in the input data. A layer of input neurons is fully connected to a 2D plane of Kohonen neurons. Each neuron is assigned a neighbourhood of local neurons. The weights between the inputs and the Kohonen layer are initialised to a uniform starting value, but with each subject to small random variations. An input vector is then presented to the network and the distance between each input and those in the Kohonen layer are measured. The shortest cumulative distance between an input vector and its neuron is the 'winner'. The winner's weights are then adjusted to make the distance even shorter. Crucially, neuronal weights in the winning neuron's neighbourhood are also adjusted in the same way, preserving the higher dimensionality of the input data. So following the training session, the SOM has organised itself to represent the input vector. The process is repeated for different input vectors, and the weights in the Kohonen layer begin to form a 2D map of the input data set.

Building an ANN

There is no well-defined method that, given a particular problem, determines the optimal architecture of an ANN. The usual approach is to start with an oversized network and use heuristic methods to 'prune' the network until the performance begins to suffer. For example, weights that are close to zero may be removed, as can nodes that have no appreciable effect on the network output or nodes that do not change their output over the entire input range. This is quite an imprecise activity, but can be improved upon by the use of meta-optimisation.

ANNs applied to the adaptive autoreclosing problem

The problem of arc detection and arc extinction are pattern recognition problems. A range of inputs exist for which the correct outputs are known, suggesting the supervised learning paradigm. Once trained, the ANN execution involves a few of simple mathematical operations, making them suitable for fast, real-time DSP algorihtms. Multi layer perceptrons are well suited to this problem and have indeed been successfully applied to the autoreclosing problem. As mentioned in chapter 2, work by Fitton [58] used an MLP to classify faulted phase waveforms into *safe to*

reclose or *not safe to reclose* boundaries. This can be visualised in figure 2.3. This is therefore the basis of the method presented in this thesis. However, there are some import improvements that will be presented in the subsequent chapter.

The optimal autoreclose is a completely separate type of problem unsuited to MLP ANN application. The optimal reclosure time takes a continuous range of values and is heavily dependent on power flows and system topology. In order to capture a useful training set, it would be necessary to simulate every possible topology with a huge range of power flows and this would be prohibitively complex in the design stage. A common criticism of ANNs is that they are unpredictable: a small change in input can often lead to a large change in output. It is therefore difficult to be completely certain about the response of an ANN when used in a real world system. As such, using ANNs to determine a value that could take a continuous range, e.g. for distance protection [80], is understandably an anathema to manufacturers and utilities. It is important to ensure the worst-case scenario, whilst mitigated, is acceptable if it comes to fruition. Only crisp Boolean outputs ensure that it is possible to know what this might be. Using ANNs in arc detection and arc extinction does fulfil these criteria for two reasons.

- The ANNs are being used to assert a Boolean value, rather than a value from a continuous range. They can only be right or wrong, not badly wrong.
- The consequences of a wrong decision in the algorithm are no worse than a failed reclose attempt using conventional autoreclosing with a fixed dead time.

5-7 Hybridised AI applications

One of the attractive things about AI techniques is their abstract nature. There are very few hard and fast rules. Depending on the application, this may also be viewed as a limitation. Either way, these methods are relatively new and they are constantly evolving within the literature. Weaknesses with a particular method may be addressed through hybridisation. This is where two or more methods are deployed in series or in parallel. Fuzzy-Expert systems have already been discussed, but there are an inexhaustible number of permutations for exploration.

In hardware, new multi-core architectures allow fast parallel computation. In power system protection this could mean that in future, a numeric relay could deploy several AI algorithms in parallel. The extent to which the algorithms agree generates a confidence value and greater inherent robustness. The confidence value also serves to show how well the secondary system understands primary system conditions. A low confidence value could be configured to trigger alarms requiring human intervention. Of all the techniques, MAS present a framework within which several other AI techniques could be deployed. One interesting area of hybridisation is meta-optimisation. This is when a primary AI system is optimised by a secondary AI technique. In power system protection, a fast online technique may be meta-optimised by a slower offline methods. For example, as mentioned earlier, there is no systematic way of finding the best MLP architecture for a particular problem. This could be addressed offline by optimising the ANN design with GAs. The considerable array of parameters defining an MLP architecture form a complex multi-dimensional search space. In such an application, the ANN architecture, number of hidden layers, transfer functions and learning algorithms are encoded as a bit string and are then selectively bred. Each member of the population is a possible ANN. The candidate ANNs are trained and tested for fitness using speed of execution and accuracy as an objective function. The most successful traits are retained in successive generations and gradually, a suitably optimised architecture emerges.

5-8 Conclusion

In this chapter, a number of AI methods have been discussed and their suitability for solving the adaptive autoreclosing problem is assessed. These are summarised in table 5.1. The three clearly defined tasks in adaptive autoreclosing suggest a parallel-hybridised technique, where a GA is used for meta-optimisation offline.

Table 5.1: Summary of autoreclosing task and the most suited AI solution

Problem	Nature of Problem	Time Critical?	Most Suitable AI Technique
Secondary Arc Detection	Pattern Recognition	Y	MLP
Secondary Arc Extinction	Pattern Recognition	Y	MLP
Optimal stability	Multidimensional Optimisation	Y	Fuzzy ES
Metaoptimisation of MPL architecture	Metaoptimisation	N	GA

In the work of Fitton [58], only the first two tasks are dealt with, and they are solved by a single ANN. The central theme of this thesis is to prove this approach is valid for systems with high penetration of wind. Optimal stability and meta-optimisation are beyond the scope of this thesis and suggested areas for further work.

Chapter 6 – Algorithm Development

6-1 Introduction

Having discussed AI techniques, it is now possible to continue with the main argument of this thesis: investigating whether adaptive autoreclosing can cope with wind generation. It is proposed that this is achieved by showing real time operation of such a scheme in a system with high penetration of wind. This chapter concentrates on the development of the autoreclosing algorithm, commencing with studies on what is possible within the context of the 132kV system. The primary arc period is found to be prohibitively similar to the permanent fault for pre-circuit breaker diagnosis. The significance of harmonic emissions are examined with reference to G5/4 engineering recommendations. The transient stability of the system is also discussed, and found not to be significant for the purposes of this work. Using this information, the last section focuses on tailoring the algorithm for the primary system model discussed in chapter 4. However, the wind farms are not included in the training cases in order to test the conclusions in Chapter 3. The algorithm is discussed in detail in three sections: arc detection, arc extinction and logic.

6-2 Autoreclosing philosophy

In [58] a single phase, adaptive autoreclosing algorithm (AdSPAR) is achieved using a single ANN. This relies on the need for single pole autoreclosing to drive the secondary arc. In the UK, the circuit breakers are three phase, and the policy is to deploy delayed autoreclosing (DAR). The UK system is interconnected enough to allow this approach: stability is not a concern and the sync check for phase drift is usually successful. The over-riding consideration is to allow time for a transient fault to clear, should it exist.

The primary goal of this thesis is testing the AdSPAR technique developed by Fitton, and to do this it must be assumed that the breakers in the Scottish system have a single pole opening facility. There are, however, two ways in which the algorithm may be adapted to use circuit breakers with only three phase breaking (AdTAR).

- Use the primary arc information, before the circuit breaker opens to diagnose the fault
- Use the secondary arc sustained by *inter circuit* coupling

In the work presented by Fitton [58], only a one phase voltage is used, belonging to the faulted phase. On a double circuit line, there are 6 phase voltages and 6 phase currents from which information can be derived. If the algorithm uses remote measurements from the substation at the far end of the line, there are a further 12 system quantities available. It was decided from the outset that remote measurements could not be relied upon for this algorithm. This was primarily because the communications link represents a single point of failure, but also latency issues coupled with a very high sample rate for high frequency measurements might pose problems. (However, communications technology improves at a phenomenal pace, so this is an interesting avenue for further enquiry. If an intelligent autoreclose scheme relied on a communication link, it could default back to conventional autoreclose if the link failed.) The current measurements were discarded since the phase shift between the voltage and the current values vary with reactive power flow. Since the proposed method is partially dependent on the time domain, by virtue of the windowed FFT, varying phase shifts would be difficult to account for in the training of the neural network. Moreover, following denerg0069sation of the isolated phases, little or no current will flow in them.

70-90% of faults are transient, and although the statistics vary with nominal transmission voltage, the vast majority of all faults are single phase to ground [4]. The algorithm is therefore only being designed to deal with single phase to ground faults. Phase to phase faults, and other rarer fault types, are not dealt with as the increased benefit to the system would not be worth the increased complexity of the scheme. Amongst other considerations, this would require modelling a different primary arc length to account for arcing between two conductors. The training set required for the neural network very quickly becomes prohibitively large. It is assumed that the main protection relay selects the correct phase and correctly diagnoses the fault type. As discussed in chapter 1, it is relatively easy to detect ground faults with the presence of

zero sequence current, and many phase selection algorithms exist in order to facilitate single pole autoreclosing.

6-3 Primary Arc

The primary arcing period yields a frequency spectrum that is very close to a permanent resistance. This is because the primary arc resistance is similar to the resistance modelled for a permanent fault. In three-phase circuit breaking, the healthy phases are also de-energised, so a secondary arc only exists if the second circuit is energised. Bo et al [29] have presented a method using high frequency transients generated before the circuit breaker to diagnose the fault type. However, this relies on a line trap, a resonant circuit in the form of capacitance and inductance in shunt with the conductors. The line trap circuit acts as a band pass filter with bandwidth of 20 kHz and a centre frequency of 50 kHz. Given that the RTDS is only capable of simulation up to 3 kHz, this approach is not possible to simulate with the system used in this work. Other methods have been put forward, but they invariably rely on specialist equipment unlikely to be available on the 132kV system.

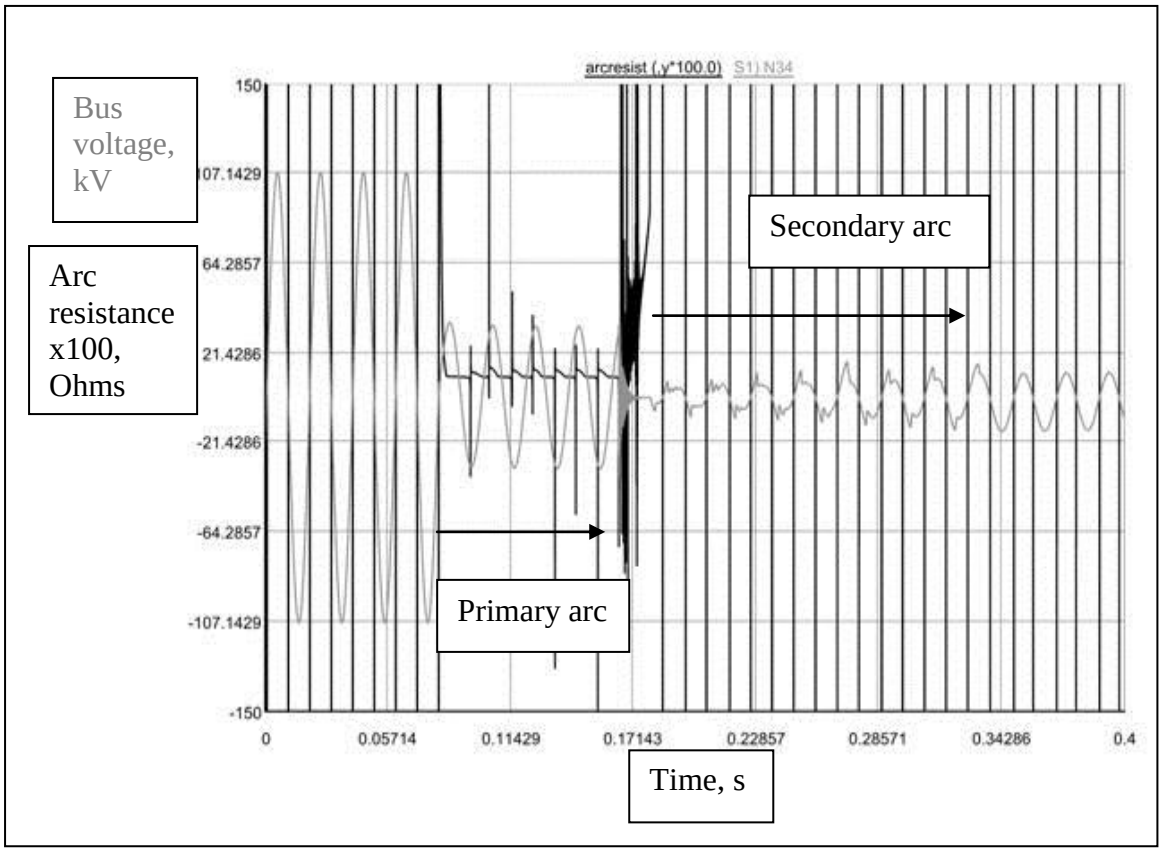


Figure 6.1: primary arc detection

Detection of the primary arc at frequencies below 3 kHz is extremely difficult because the resistance is fairly constant and of a similar magnitude to a fault modelled by a small permanent resistance, in the order of 1 Ω. Figure 6.1 shows the primary arc resistance (in Ω x 100 for scale) plotted against the sending bus voltage. (The resistance is obtained by dividing the instantaneous fault bus node voltage by the arc current from the model. Consequently, when the current passes through a zero there is a spike to represent the divide overflow - the simulation emits a value of 1 $x10^8$ Ω. However, the information pre circuit breaker should be discarded since the arc model is not connected in this period. In the secondary arc period, the resistance varies off the scale in each cycle, giving rise to the distinctive secondary arc signature). In the primary arcing period, it can be seen the arc resistance is fairly constant and close to zero.

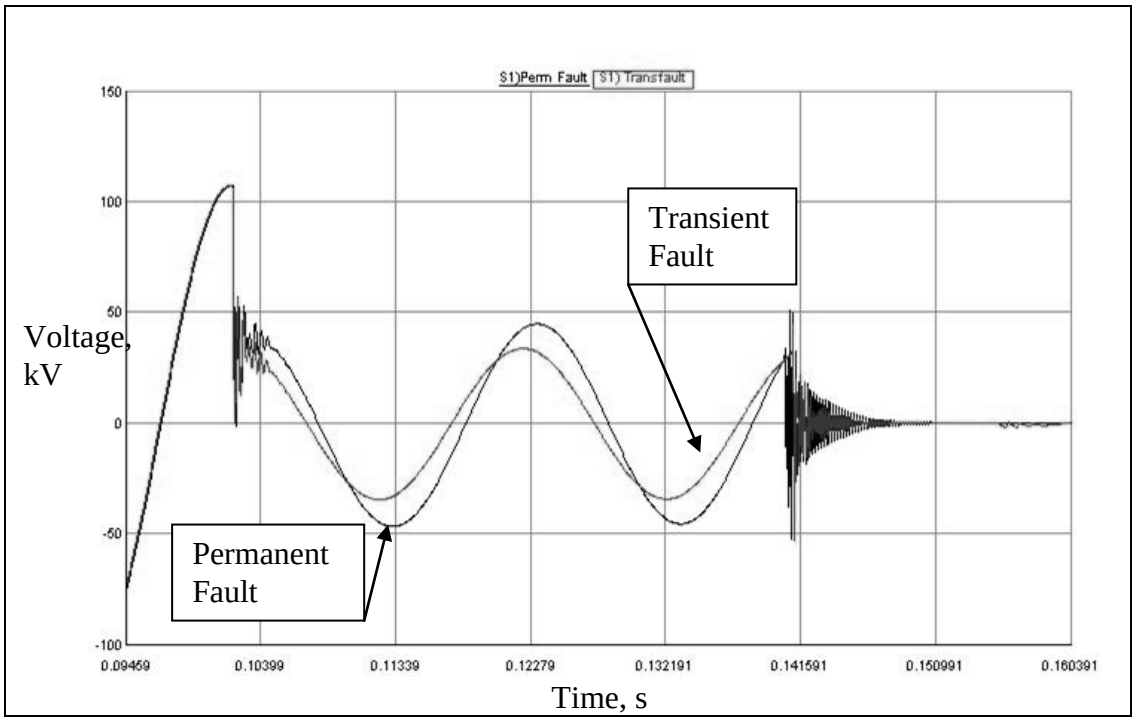


Figure 6.2: Transient and permanent post-fault, pre breaker waveforms

This is observed in the similarity between permanent and transient arc waveforms, both plotted in figure 6.2. The fourier transform of the post fault, pre circuit breaker operation, was computed on these signatures and is shown in figure 6.3 between 0 and 2 kHz. Apart from the *magnitude* of the fundamental, best viewed on 6.2, the resulting spectra are very similar. Using the fundamental voltage is not robust because it varies

more significantly with other parameters such as fault resistance and location on the line.

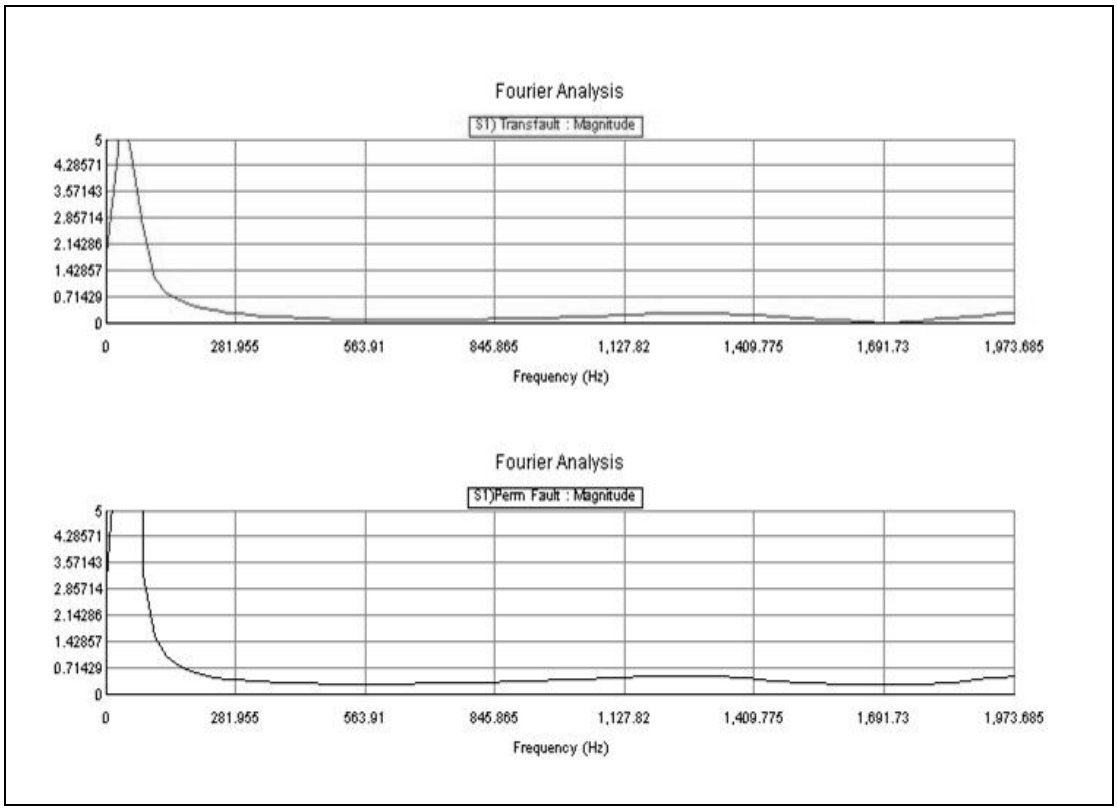


Figure 6.3: frequency spectra of primary arc and permanent resistance

Extensive tests show that the healthy voltage phases offer even less variation between permanent and transient faults. Moreover, the frequency spectrum across all phases show greater variation in fault location than fault type. When the fault is remote from the relay, the proportion of resistance due the fault is less than the resistance of the line (around 4 Ω for the 30 km Beauly-Farr line). For brevity, these figures are omitted. To be absolutely certain there is no potential in using the primary arcing period, trials were conducted using a MLP ANN. However, the trained ANN responds poorly, as it is not able to recognise primary arcing in the test waveforms. At the ANN input stage, the changes due to fault type were evidently masked by the much greater variance in the other parameters. This explains the absence of primary arc detection methods in the literature.

It was decided on this basis that if the algorithm was to be adapted to three pole circuit breaking, it must rely on the secondary arc due to *inter-circuit* coupling. This is investigated in Chapter 7.

6-4 Harmonic emissions

Chapter 3 of this thesis documents a study on the effect of wind generation on EM transients in the case of short circuit faults. The general conclusion was that whilst there were some differences in the generating technology, the variation in the other parameters that affect fault waveforms were far more significant. The main concern over wind farm technology and ANN based adaptive autoreclosing is the harmonic emissions caused by power electronic converters. In order to specifically address this within the RTDS model, a study was undertaken examining harmonic pollution. This study is also helpful because it includes a realistic arc model that was not used in the chapter 3 investigation.

Any generator in the UK must comply to operating constraints imposed by the utility, known collectively as the *grid code*. An important aspect of this is ensuring a certain degree of power quality. The UK grid code demands all generators to be compliant to Engineering Recommendation G 5/4, which itself is based on the international standard IEC 61000 [81]. This governs the levels of harmonic distortion permissible, up to and including the 50^{th} harmonic. ER G 5/4 recommends that harmonic voltages in systems between 20kV and 145kV do exceeded the levels in table 6.1.

Table 6.1: G 5/4 maximum levels for the first 9 harmonics on systems between 20kV and 145kV

Harmonic Number (h)	Harmonic Voltage %	Harmonic Number (h)	Harmonic Voltage %
2	1	6	0.5
3	2	7	2
4	0.8	8	0.4
5	2	9	1

In practice, wind farms have filters to suppress the level of harmonic distortion and the operational levels should be well below the permitted levels. However, as a worst

case, harmonic emissions should not exceed those stated in G 5/4. The harmonic source component in RSCAD allows up to four harmonics superimposed on the fundamental AC sinusoidal source. The magnitudes of these harmonics are expressed as a percentage of the fundamental. If the source voltage is increased in run time, then the harmonics also increase in proportion to the fundamental. So for this investigation, the wind farm models were removed, and the farms were modelled as harmonic voltage sources with behind an equivalent subtransient impedance, derived from the short circuit capacity at their point of connection. Harmonic emissions equivalent to the maximum permitted G 5/4 levels represented both wind farms. It has been observed with PWM back-to-back converters, the 5^{th} and 7^{th} harmonics are most significant occurring at 250 Hz and 350 Hz respectively [82]. The four harmonics were chosen to be the 3^{rd}, 5^{th}, 7^{th} and 9^{th} with respective harmonic voltage levels of 2%, 2%, 2% and 1%, as per G 5/4 levels expressed in table 6.1.

Spectrographic analysis

Figures 6.4 – 6.11 are 3D spectrograms showing a logarithmic plot of time against frequency and magnitude in decibels. In order to obtain the decibel value, the voltages were converted to power with a load resistance of 1 Ω. Using decibel values makes it easier to observe subtle differences in the spectrograms at high frequencies.

The pure time domain voltage is also plotted on each graph for reference, but this plot is laid flat at magnitude 0 dB to differentiate it from the spectrogram. The figures 6.4 and 6.5 show the control cases and are annotated with respect to the time axis to show the events of interest. The primary arcing period are the two swells running parallel to the frequency axis and are shown in figure 6.4 between points 1 and 2. These are the fault inception and the circuit breaker operation. As a step change in circuit conditions, these events manifest themselves over all frequencies but are localised in time. As mentioned in the previous section, the primary arcing period is not relevant, since it is not feasible to make a diagnosis between a transient and a permanent fault in this region. Further confirmation of this can be seen when the period between points 1 and 2 on figure 6.5, is compared from the same section in figure 6.4, and appears to be identical.

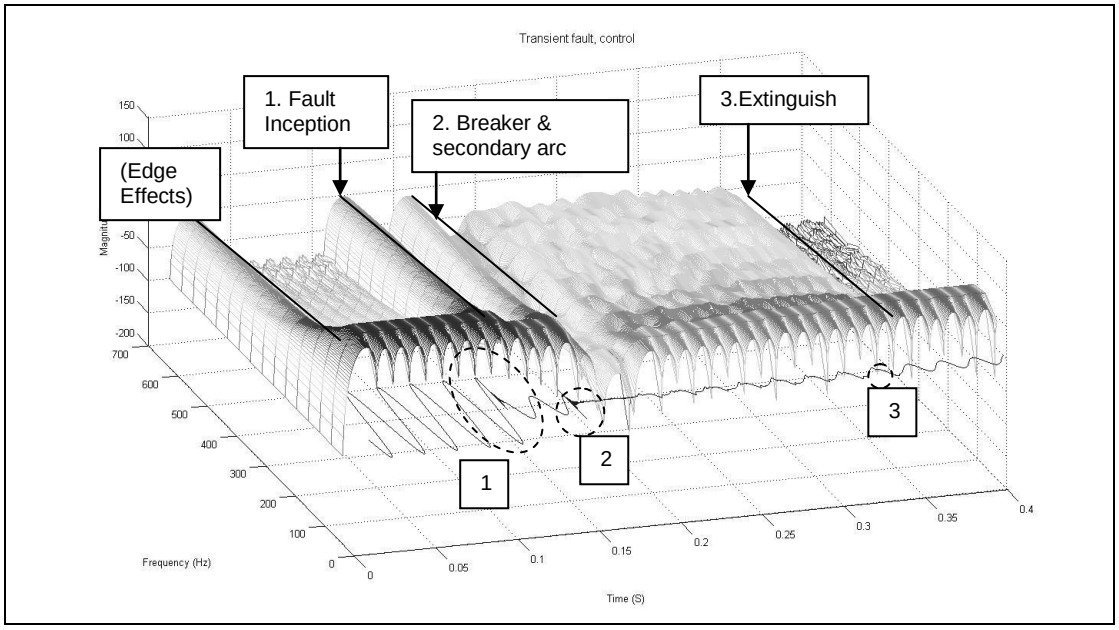


Figure 6.4: Spectrogram plot of transient fault no harmonics or CVT

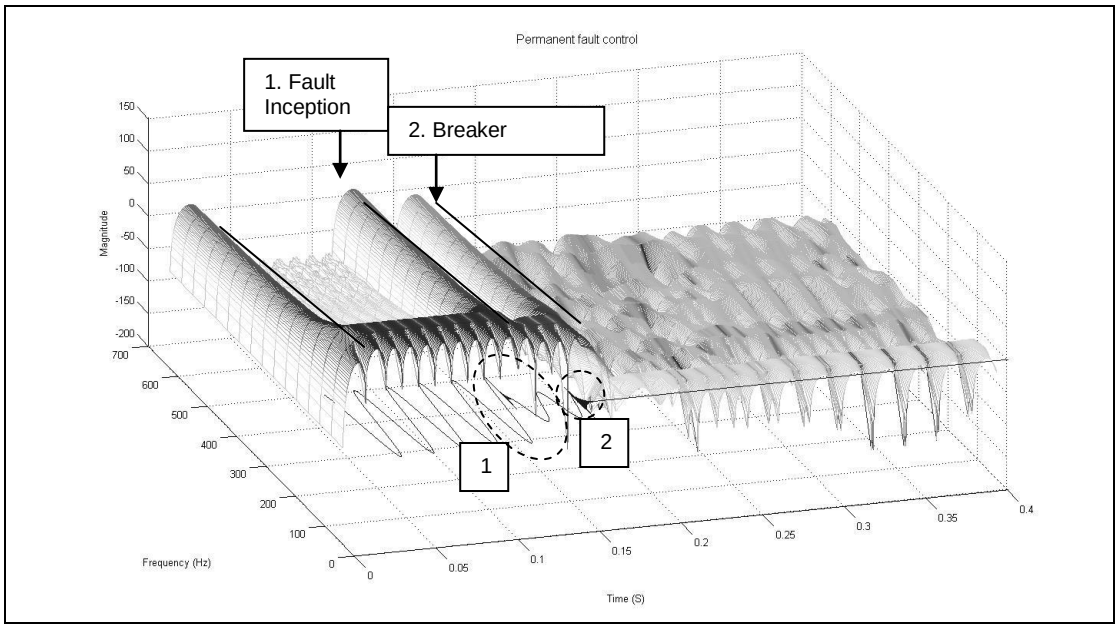


Figure 6.5: Spectrogram plot of permanent fault no harmonics or CVT

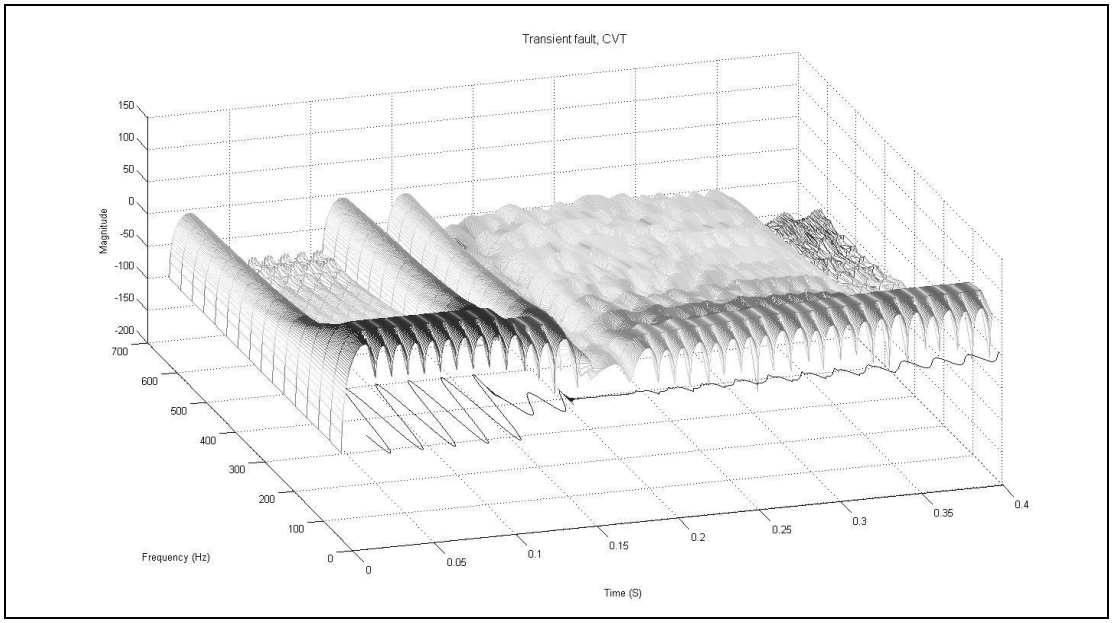


Figure 6.6: Spectrogram plot of transient fault with CVT

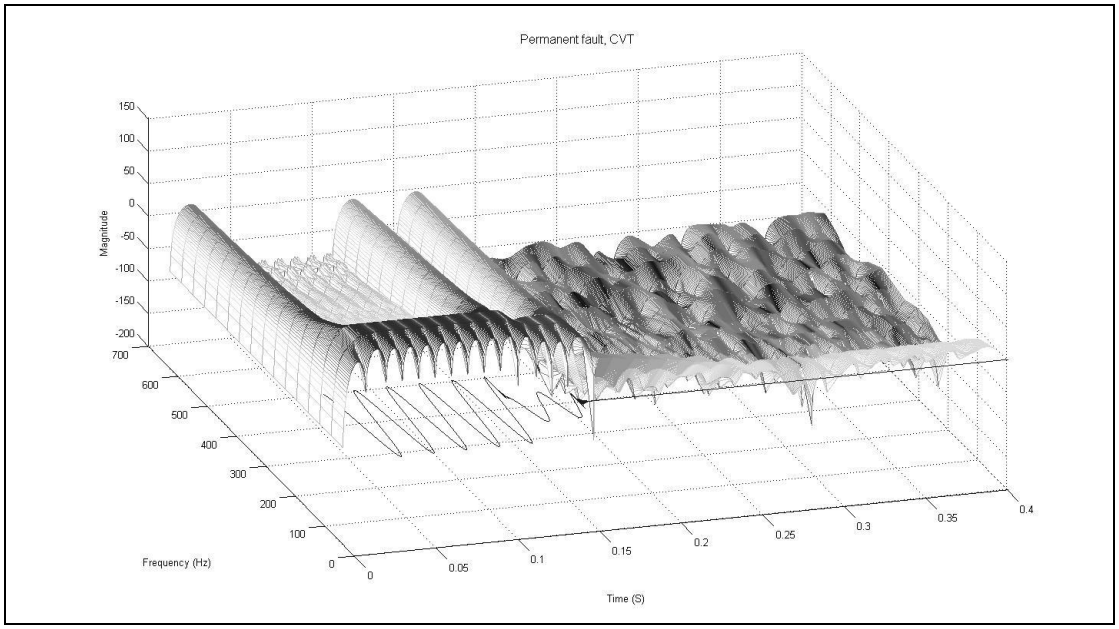


Figure 6.7: Spectrogram plot of permanent fault with CVT

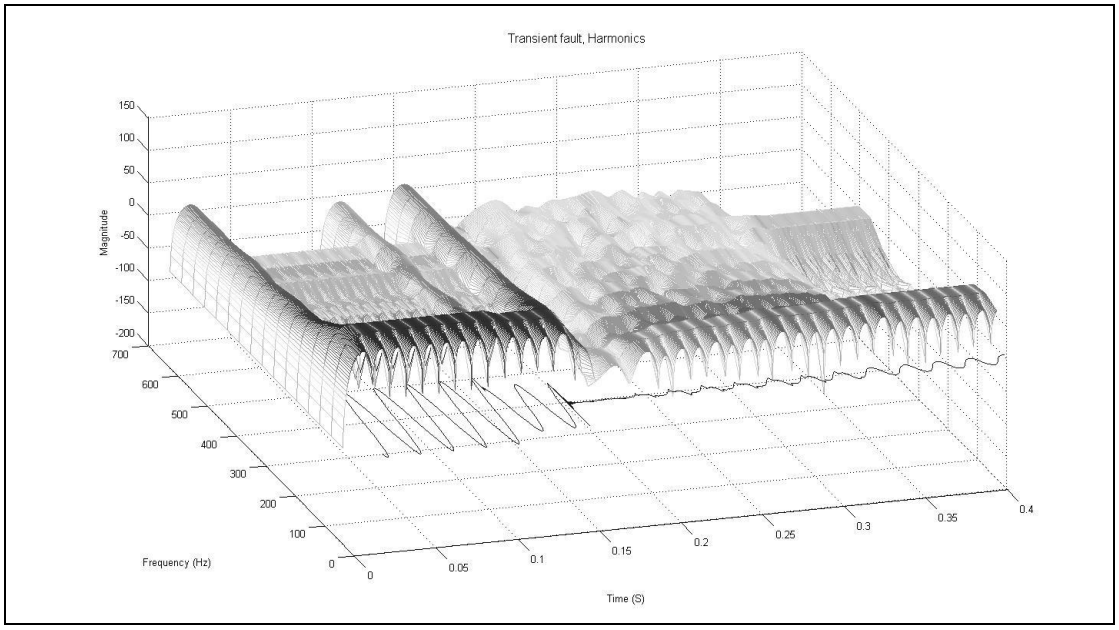


Figure 6.8: Spectrogram plot of transient fault with harmonics

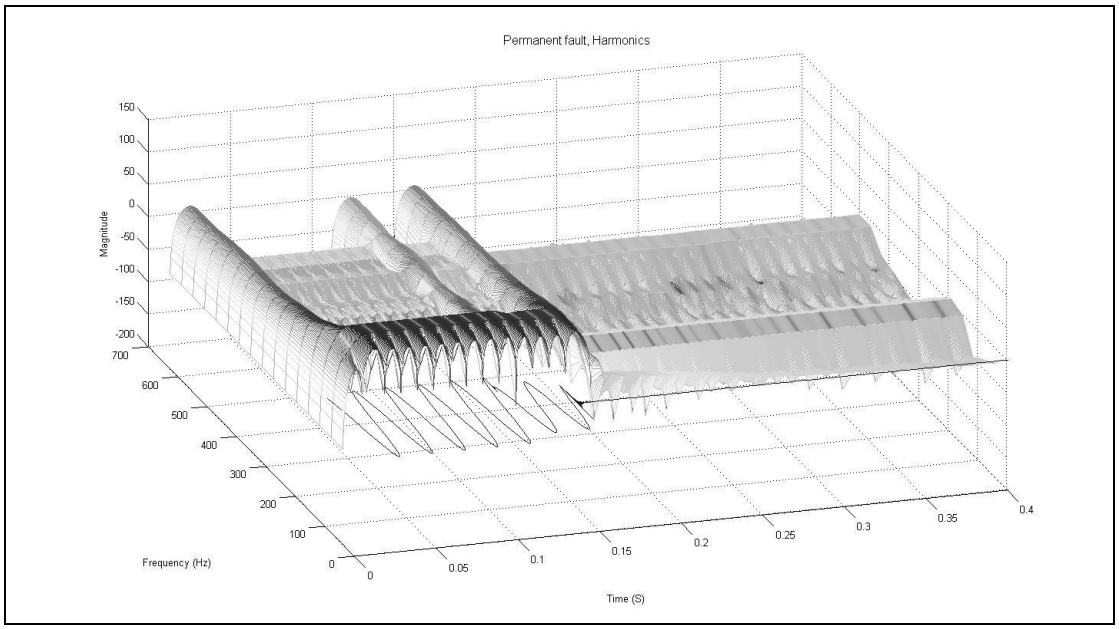


Figure 6.9: Spectrogram plot of permanent fault with harmonics

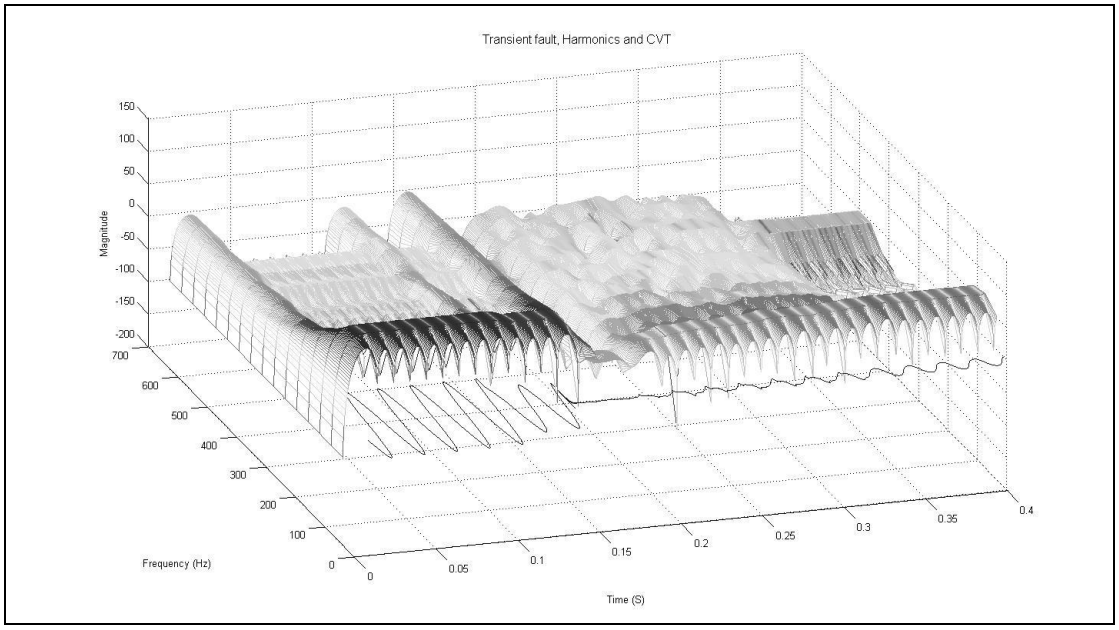


Figure 6.10: Spectrogram plot of transient fault with harmonics and CVT

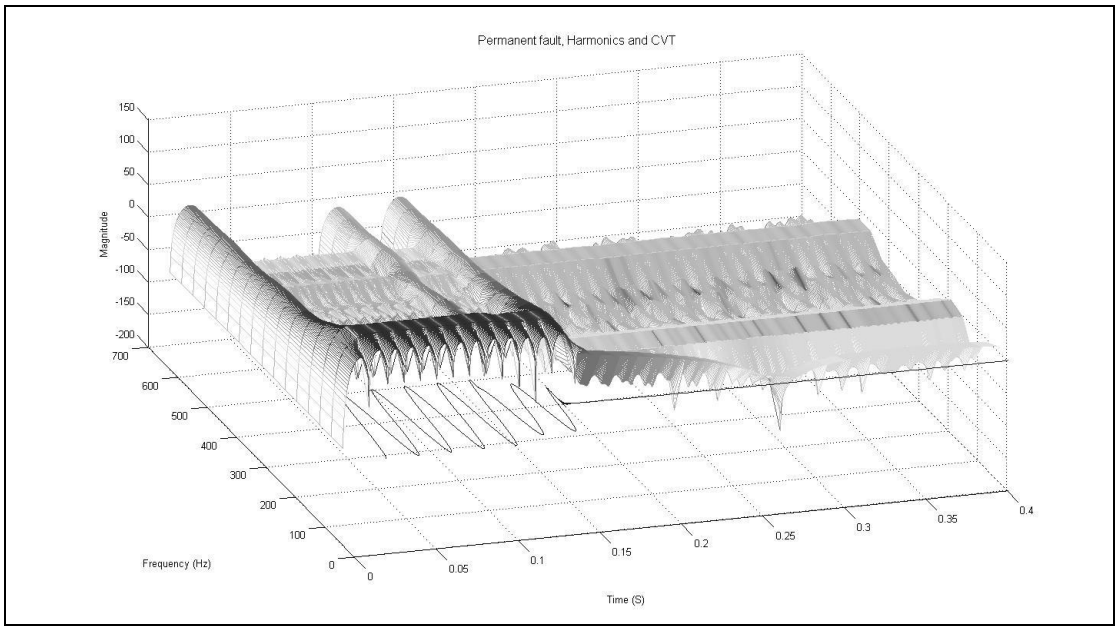
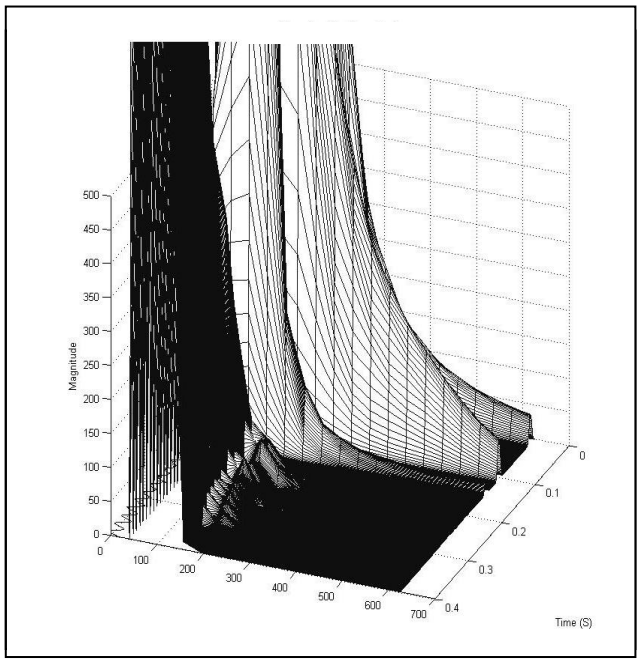
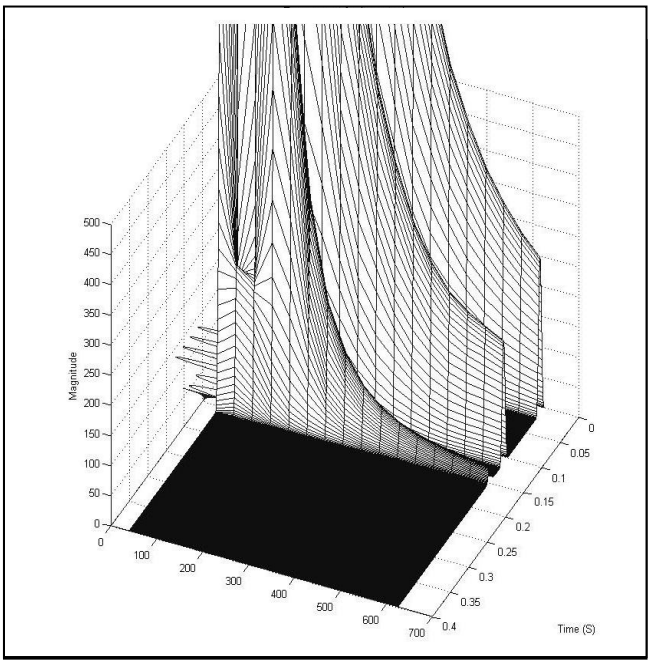


Figure 6.11: spectrogram plot of permanent fault with harmonics and CVT

It can be seen that the harmonics cause a noticeable, albeit uniform, distortion of the spectrogram centred around their harmonic frequency. These are expressed as swells in the spectrogram running parallel with the time axis, observable in figures 6.8 – 6.11. However, these are around 10 dB less than the high frequency energy that is

present all across the spectrum in the case of a transient fault, shown in figures 6.4, 6.6, 6.8 and 6.10. The harmonic swells are more noticeable for a permanent fault because there is no high frequency arcing pollution. If the sensitivity of the ANN can be calibrated so that it only responds to the high arcing energy, the autoreclosing scheme will not be affected by this level of harmonic pollution. The CVT cases are included for comparison to show their effect on the waveform. Again, any influence due to the CVT on the high frequency spectrum are well below the magnitude of the transient arcing faults. It is only in the case of the permanent fault, or when the arcing fault has extinguished, that CVT or harmonics become apparent in the spectrogram. In order to suppress these differences, the decibel conversion was omitted during the feature selection process. Since the scale ceases to be logarithmic, CVT and harmonic pollution effects are far less significant. Spectrograms plotted from an angle that makes the post breaker operation high frequencies easier to observe is shown in figure 6.12 (transient) and 6.13 (permanent). With the decibel calculation removed, only the information above 300 Hz is particularly significant.





Above left: Figure 6.12: Reverse spectrogram plot of transient and,
above right: Figure 6.13: Permanent fault,
with harmonics and CVT

Separate from the effects of harmonics and CVTs, the 50 Hz component in the permanent fault in this system is generally much less than the 50 Hz component when the transient fault has extinguished. This is due to the shorter line length than is typical at supergrid voltage, and the result being less coupling with the healthy phases. In the permanent fault case, the magnitude of the 50 Hz component is highly dependent on where the fault is on the line. In the case of the transient fault, when the arc has extinguished, a current path to ground no longer exists, so mutual coupling exists along the entire line length regardless of where the fault occurs. This means it is important to train the ANN for faults in different locations, especially for permanent faults.

6-5 Transient stability of 132kV system

Power flow between synchronous machines is determined by the angular difference of each of the machine's rotor angles. This relationship is highly nonlinear - increasingly so for a real world multi-machine system each with separate controls. For simplicity, consider a two-machine system connected via a transmission line. The two machines, at respective voltages V_1 and V_2 have internal reactances X_1 and X_2. The transmission line has an equivalent inductive reactance but negligible capacitance and resistance. At transmission level voltage, X>>R. In this case, machine 1 is acting as a generator and machine 2 is acting as a motor, and power is being transferred from machine 1 to machine 2. A single line diagram is shown in figure 6.14.

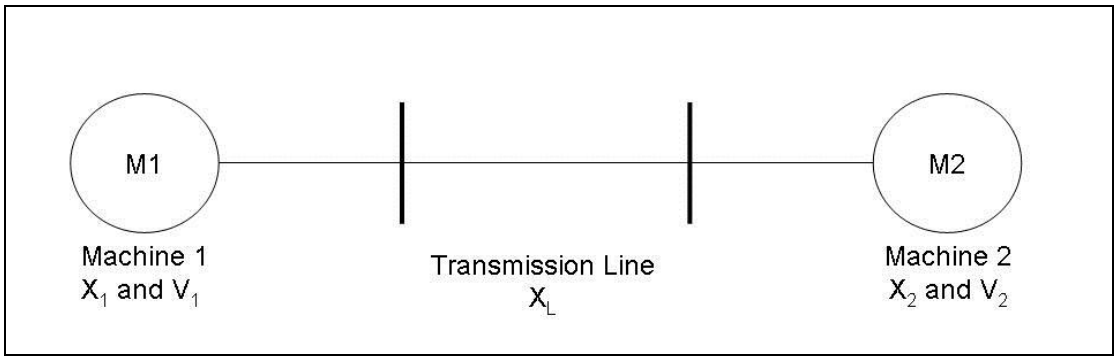


Figure 6.14: Two machine system

Assuming ideal machine characteristics, the total reactance of the system is given by (6.1).

$$X_T = X_1 + X_L + X_2 \quad (6.1)$$

The power transferred between machine 1 and machine 2 is described by the fundamental equations (6.2) and (6.3).

$$\Delta P = \frac{V_1 V_2}{X_T} \sin\delta \quad (6.2)$$

$$\Delta P_{\max} = \frac{V_1 V_2}{X_T} \quad (6.3)$$

Where ΔP is the power transfer and δ is the angular difference between the rotor angles of the two generators. Thus keeping all other values constant, the relationship between power transfer and δ is sinusoidal, sketched in figure 6.15.

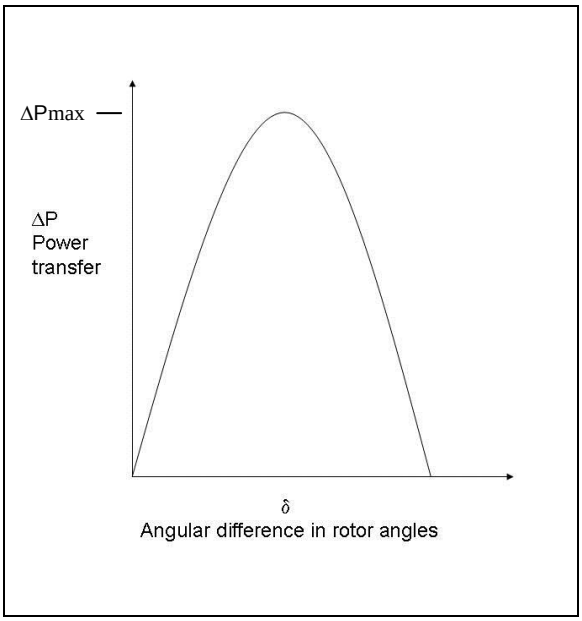


Figure 6.15: Power transfer dependence on angle difference

Beyond a certain value of δ, the two generators will lose synchronism with each other. Beyond ΔP_{max} an increase in the rotor angle results in a *decrease* in power transfer. In other words, the forces acting on the rotor angles are no longer restorative to an equilibrium condition and thus synchronism collapses. In the case of a two machine model, this value of δ is 90 degrees.

'Transient stability' is the ability of the system to retain synchronism after large disturbances such as a fault. The assumption is that stability is lost after the first power swing. (However, this may not be the case, especially when autoreclosing is applied as it represents a secondary step change in system conditions [55].)

Although the RTDS model is far more complicated than a two-machine model, there are several reasons why transient instability is not a concern in this system.

- On short lines, such as the 30 km on the Beauly–Farr line, thermal limits dictate the maximum power transfer
- The system is more interconnected, via the ring topology, than a two-bus system. There is therefore an alternative route for the synchronising power to travel, increasing stability.
- A three phase fault is the more onerous than a single phase to ground in terms of stability – in this case the autoreclosing algorithm is not applied as the fault is assumed to be permanent

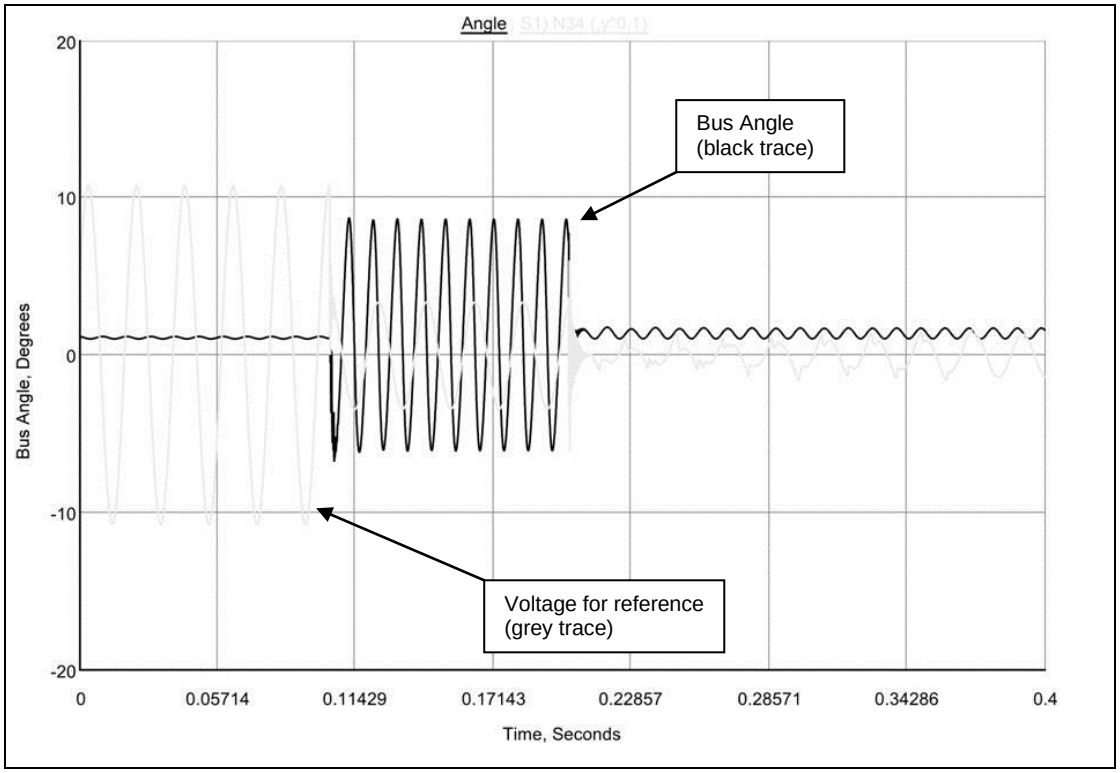


Figure 6.16: Transmission angle against single phase transient fault

With reference to chapter 4, the power flow through the system was designed to be the peak winter flow for as quoted in National Grid's 7-year statement. Figure 6.16 shows the angle difference between the Beauly bus and the Farr bus plotted for a single phase to ground transient fault. (The faulted phase voltage is also plotted as a lighter trace for time reference, but this is not to scale). It can be seen that the system is stable even if the circuit breaker operation is extended to 8 cycles, as in the case shown. Transient oscillations caused by the single-phase circuit breaker attenuate very quickly. However, some oscillation following the circuit breaker persists owing to the imbalance in the system.

6-6 Neural network task separation

The previous chapter highlighted several potential uses of AI in the autoreclosing algorithm. The earlier sections of this chapter further validates the ANN approach developed by Fitton et al [58]. The presence of harmonic pollution and the effects of the CVT introduce further complexity in the pattern recognition problem. The ANN is well suited to this problem because the designer does not have to understand the

complexity of the problem, only train the ANN with carefully chosen training cases so it can generalise, exhibiting the desired behaviour for cases it has not seen before. This said, in protection and control applications, the designer must make the algorithm as robust as is reasonably possible. As mentioned earlier, pattern recognition is only a Boolean outcome, and the consequences of a wrong decision are no worse than conventional autoreclosure.

In the Fitton method, a single ANN is used to indicate the 'safe to reclose' condition. In other words, the tasks of secondary arc detection and secondary arc extinction are performed by a single network. This approach sacrifices robustness for conceptual simplicity and computational efficiency. It is also a good showcase of the ANN's ability to recognise patterns in complex problem spaces. However, the overriding consideration in protection should be on robustness. For a permanent fault, the sinusoidal oscillation caused by the coupling of the phases is similar to the oscillation once the arc has extinguished, since in both fault cases there is little high frequency content. However, these two scenarios belong to different cases, as it is not desirable to reclose if a permanent fault exists, but reclosure should be sanctioned if, in the case of a transient fault, the arc has extinguished. This is shown in figure 6.17.

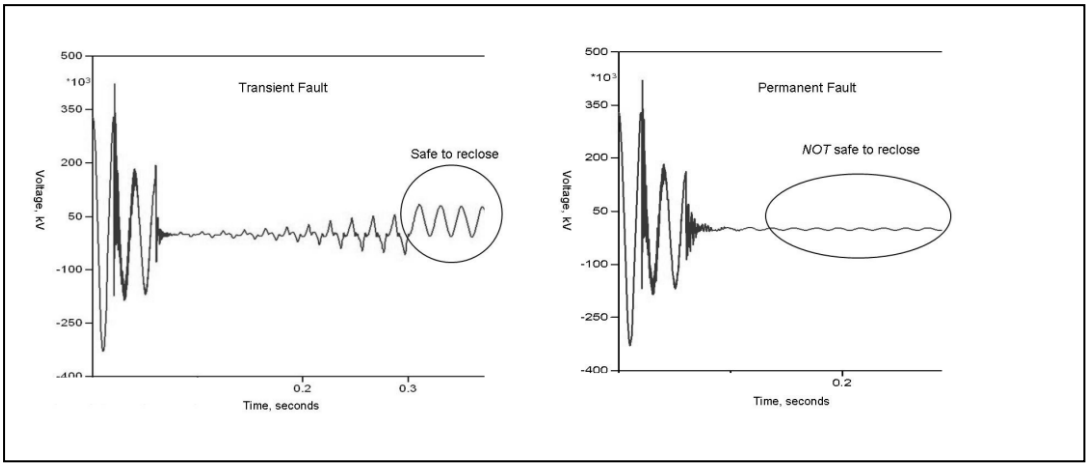


Figure 6.17: The distinction between safe and not safe to reclose

Therefore, a separate MLP was assigned to each key task: the first was trained to recognise the presence of an arcing fault. The second was trained to look for arc extinction. The logic issues a reclose decision if, and only if, the first MLP had diagnosed the fault to be transient.

Algorithm Overview

The ANN was developed using Matlab's neural network toolbox. Waveforms recorded by the RTDS simulation where used in ANN training offline. The ANNs were exported to simulink, where the autoreclosing logic was developed. The next chapter will discuss the hardware and software tools used for real time verification of the algorithm. An overview of the scheme is shown in figure 6.18 as viewed in Matlab's simulink.

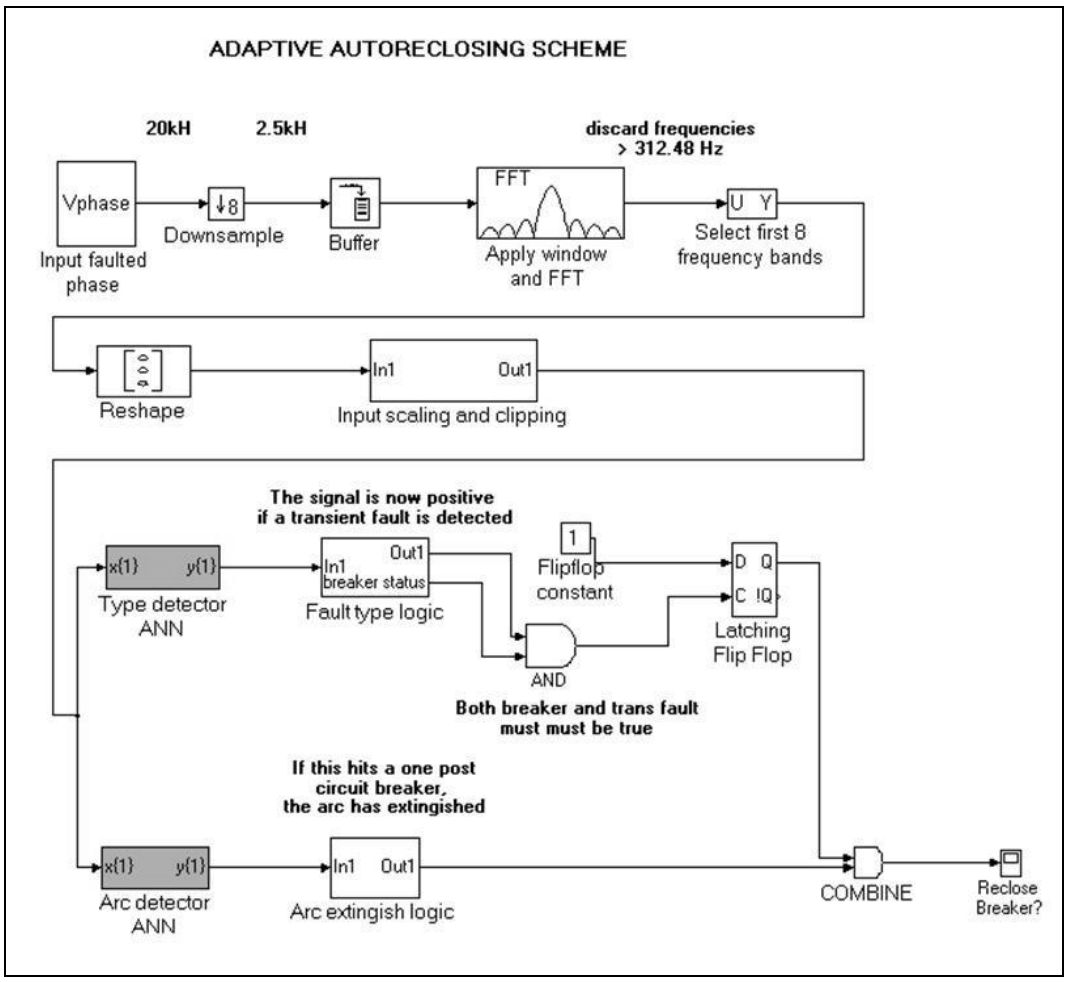


Figure 6.18: The algorithm visualised in Simulink blocks

6-7 ANN training cases

Given that only the period following the circuit breaker operation is relevant to the algorithm, the training waveforms were cut to the secondary arcing period. As discussed in chapter 3, there are certain parameters that affect the faulted transient response of the power system:

- fault inception point on wave
- nature of the fault (arcing or permanent)
- permanent fault path resistance
- source capacities
- location of the fault point on the line
- line length
- nominal system voltage

The investigation concluded that the generating technology used had some bearing on the fault signatures, but these were less important than other parameters, and significantly, those that were unknown prefault. In order to test this conclusion, it would be necessary to test an ANN based AA scheme on a system with wind technology. Importantly, the networks used in this scheme should not be trained with input waveforms generated with the wind farm model because the ANN would only be memorising rather than learning to generalise. Thus the training cases were chosen to reflect all the significant factors except the generating technology. Matlab's resilient back propagation algorithm was used in network training.

Fault inception point

Three different fault inception points were used, at voltage maxima, voltage zero and at voltage midpoint. Ostensibly, fault inception point is not relevant since secondary arcing occurs after circuit breaker operation rather than the initial fault. In practice, H.V. circuit breakers can only interrupt near a current zero crossing. However, *voltage* inception point varies depending on reactive power flow through the line and concomitant phase shift between current and voltage. Assuming a constant response time of exactly 5 cycles following the inception angle delay yielded a the same variation in the secondary arc inception time. This was simpler than altering reactive power flow and setting up a current zero crossing detector in the circuit breaker operation. It should be mentioned that the RTDS circuit breaker is modelled as an ideal switch. In practice, the arc quenching operation within the circuit breaker should be taken into account in the simulation. Fortunately, the secondary arc persists over many cycles, whereas the circuit breaker arc transients attenuate very rapidly. The

output of the neural networks was therefore not taken into account until a full cycle after circuit breaker operation, in order to allow transients to attenuate.

Fault Type

This was the *raison d'etre* for training the networks. In the case of a transient fault, the RTDS arcing fault model, detailed in chapter 4, was used. For a permanent fault, a constant resistance, of default value 2 Ω was used. For the fault type detector ANN, the fault type formed the target vectors in the training. (In other words, the output neuron was trained to emit a 1 or zero depending on fault type).

Fault Resistance

The fault resistance in the permanent fault case depends on how it is caused. Technically, there is no upper limit but high impedance faults often fail to trip the initial protection relay so are not within the scope of this work. Based on previous papers, the fault resistance was varied in the following discrete steps at 0, 2, 50 and 100 Ω.

Fault location

This is perhaps the most important of all the parameters because it determines the amount of additional impedance seen by the relay, so it affects both transient and arcing faults. However, the amount of training cases achievable with the travelling wave line model is limited since no line section can be below 15 km. Faults were therefore trained in 3 locations, 100% 50% and 0. On the Beauly-Farr line this corresponds to location 30.1 km, 15.05 km and 0 km from the sending bus. In the 0 km case the arc fault was at the same node as the relay.

Source Capacities, Line Length and System Voltage

The source short circuit capacities, line length and system voltage were not varied as they are fixed properties of the chosen section of Scottish network.

Generation Type

Lastly, and most importantly, the generation type was not changed. The wind farm models were omitted for the training cases and were instead modelled as RTDS

voltage sources, their subtransient reactance calculated from their short circuit capacities. The S.C.C. of the Farr and Paul's Hill wind farm was not available at the time of writing, and so was assumed to be 3.5 GVA. Although this may in reality be somewhat inaccurate, it does demonstrate the robustness of the ANNs since their performance is later tested on the actual wind farm model with the correct capacity.

6- 8 Downsampling

The sampling rate of the RTDS is 20 kHz, the inverse of the fixed timestep of 50μs. RTDS quote an upper fidelity limit of 3 kHz, so all frequencies beyond this level should be discarded. The incoming data was therefore first downsampled to 2.5 kHz. This also makes Fourier transformation much faster. The downsampling simply involved discarding seven samples in every eight.

6-9 Feature extraction

Although some ANN-based adaptive autoreclosing algorithms have made use of Wavelet transforms (see chapter 3), it was decided, based on conclusions made in [21], to use Fourier transformation in the feature extraction process. This MSc dissertation compared the short time fast fourier transform (STFFT) with the discrete wavelet transform (DWT) and found the former to be favourable when used for the Fitton ANN approach. In addition, the Fitton method is the only scheme that has been documented on a real world system, and this uses the STFFT. Extraction of the features in the time series is therefore performed by the FFT of a moving window. The FFT is a more computationally efficient version of the DFT. If x is a vector of length n, the one dimensional discrete fourier transform yields a vector also of length n, and is described by equation (6.4).

$$y_{p+1} = \sum_{j=0}^{n-1} (e^{-2\Pi i/n})^{jp} x_{j+1} \qquad (6.4)$$

Where p is the index of the vector y and j is the index of vector x. Both p and j run from 0 to n-1. If x is a continuously sampled discrete time series or space vector, y is

defined as the frequencies within such a signal between DC and the sampling rate Fs. The complex values give the magnitude and the phase shift at *j* common frequency intervals [83]. (In (6.4), *i* not *j*, is the imaginary square root of minus 1.) In order to preserve time domain information, the transform must take place over a moving window. In the feature selection stage, the incoming time series fills a buffer, the buffer undergoes a Hann windowing function (to minimise spectral bias), the FFT of this window is computed, and then the buffer is moved on by one sample and the process is repeated. The buffer is of length 64 samples. Only the magnitude of the frequency components is used.

The Hann window takes the form of (6.5).

$$w(n) = 0.5\left(1.5 - \cos\left(2\Pi\frac{n}{N}\right)\right),\ 0 \le n \le N \qquad (6.5)$$

Where L is of length (6.6).

$$L = N + 1 \qquad (6.6)$$

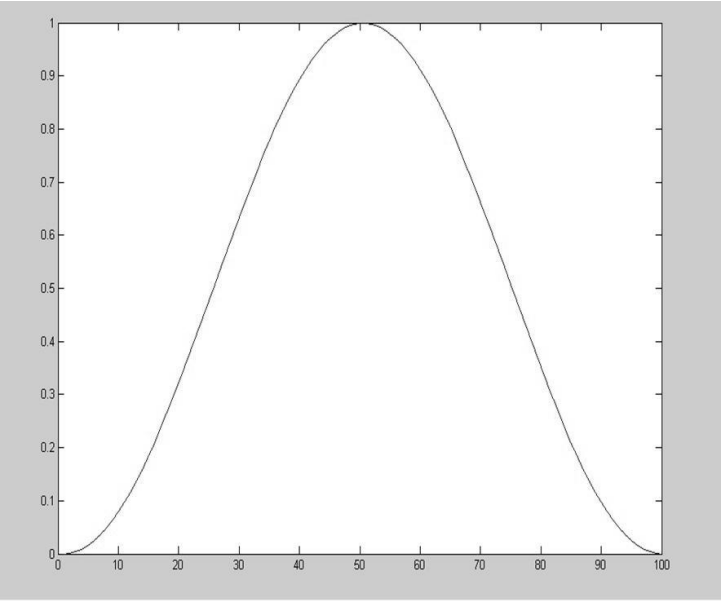


Figure 6.19: Hann window function

An example Hann window is shown in figure 6.19. The length of the window is defined by the size of the buffer it is fed from in this algorithm.

For maximum efficiency, the FFT requires a window of length that is a power of 2. Therefore, $2^5 = 64$ gives the best compromise between speed and frequency resolution, yielding frequency bands 39.06 Hz wide. (If, for example, an FFT of length 32 was chosen, this would not cover the fundamental power system component of 50 Hz). The result of the STFFT is a series of time dependent frequency bands, describing the frequency content of the incoming time series up to 2.5 kHz. Plotting this information in three dimensions yields a spectrogram similar to those shown in section 6.3.

Since this algorithm assumes typical measurement transducers, frequencies above 700 Hz cannot be used due to the roll off in the frequency response of the CVT. Figures 6.12 and 6.13 in section 6.4 show the isometric spectrogram plots viewed from a different angle with the decibel conversion omitted. It can be observed that there is very little information above 300 Hz in either a transient or a permanent fault. Therefore, only the first eight frequency bands from the FFT were presented to the ANN. These represent information from the frequency spectrum up to 312.48 Hz

6-10 Normalisation

A brief but important pre-processing stage is the normalisation of the input data. This is necessary since the 50 Hz component is much greater than the high frequency components, and this tends to dwarf the higher frequency bands. Moreover, it is far more robust from an ANN perspective to have inputs clipped between 1 and -1. The arc is a non-linear and complex phenomenon. If an anomalous frequency event occurs that has not been covered in the training data, the network may not respond appropriately. At this stage the input data is normalised between 1 and -1 and clipped to 1 should it exceed the maximum, and -1 should it exceed the minimum. The X_{max} and X_{min} were selected to be the maximum and minimum magnitudes encountered in each specific frequency band averaged for all the training waveforms. The transfer function may be summarised by equations 6.7 – 6.9.

$$Y_{out} = f(x) = x\left[\frac{2}{X_{\max} - X_{\min}}\right]\text{-}1 \quad (6.7)$$

Although for x > Xmax

$$Y_{out} = 1 \quad (6.8)$$

And for x < Xmin

$$Y_{out} = -1 \quad (6.9)$$

6-11 Architecture of Neural Networks

Two multi-layer perceptrons were used, one to determine the fault type and one to detect secondary arc extinction. Clearly, information from the second neural network is only relevant if the first network deems the fault to be transient. If the fault is deemed to be permanent, there is no arc extinguish to diagnose and autoreclosure is blocked. The logic blocks used to implement this are shown in figure 6.16.

The input and output layers in the neural networks are fixed by the task in hand. There are eight inputs to the neural network, as these form equally sized frequency boundaries up to 312.48 Hz.

No systematic method exists to determine the optimal hidden layer size. Since they are used to fit patterns to information based on pre-existing data, there is no exhaustive way of testing the performance of one given architecture against another. After all, if every case that they are to encounter is available, there is usually a more deterministic way to separate the data set into the classes. Moreover, an inherent quality of ANNs is their robustness in that they are tolerant to failure of one or more processing units. Several *rules of thumb* do exist, but these are highly dependent on the problem to which the network is applied. Usually, heuristic methods are used to prune an oversized network into a smaller, more efficient one. This may involve removing branches that have little or no significance due to weighting, or removing

neurons which exhibit little activity. However in the author's experience, within wide limits, the hidden layer size of the ANN is fairly arbitrary for this application, in that small and large layers perform equally well. In some cases, an overly large number of hidden neurons will cause the ANN's ability to generalise to suffer, but the upper limit in this application was determined by initial concerns over the real time computational load. Based on work presented in [58] and trial and error manipulation, both of the neural network sizes were selected to be a single layer of 12 neurons.

6-12 Time domain response

In each time step, the frequency bands at that window are fed to the input layer of the neural network and the output is computed. Each network is trained offline to classify a single time step as belonging to one class or another. The typical time domain response of the neural networks is shown in figures 6.20 and 6.21.

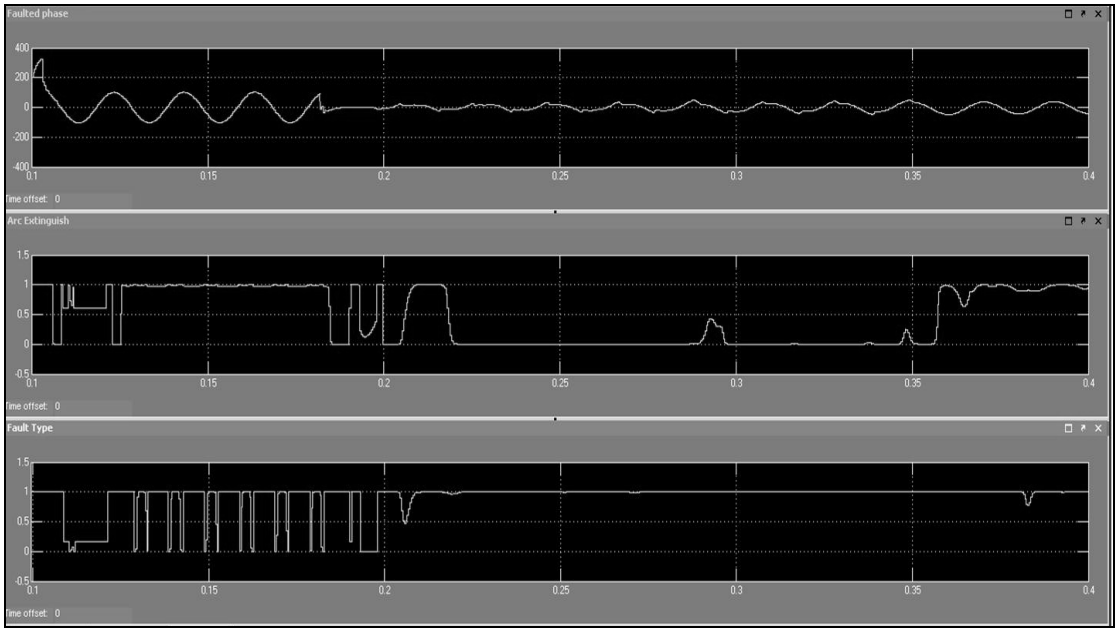


Figure 6.20: Time domain ANN responses to transient fault. Top trace: secondary arc CVT, middle trace: Arc Extinguish Detector, bottom trace: Arc Type Detector

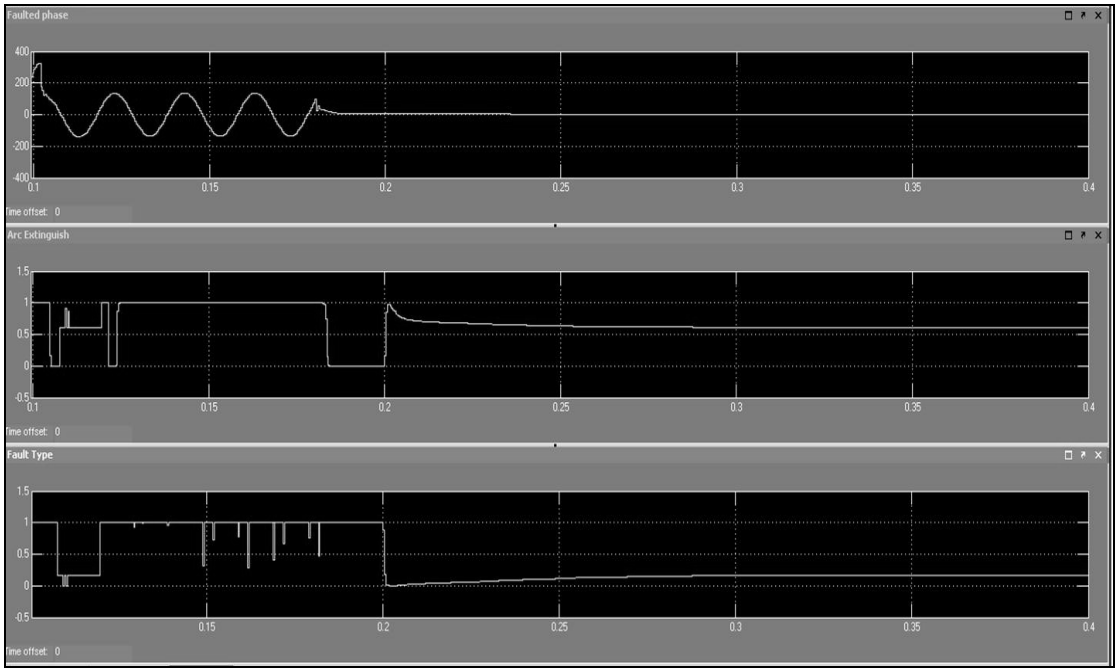


Figure 6.21: Time domain ANN response of permanent fault. Top trace: permanent fault CVT, middle trace: Arc Extinguish Detector, bottom trace: Arc Type Detector

The networks are trained to respond with 0 or 1 appropriate to its specific task. However, output is rarely precisely 0 or 1 and can lie anywhere between these values. For this reason, a time domain response of greater than 0.5 is rounded to indicate one and less than 0.5 indicates a zero. In addition, information from both networks is ignored for a whole cycle post circuit breaker to allow the circuit breaker opening transients to attenuate. With these caveats in mind, the response of the type detecting neural network (third trace, figures 6.20 and 6.21) correctly diagnoses each fault case. In the case where the secondary arc is present, the extinguish is correctly diagnosed when the response of the network swings from 0 to 1 as seen in the first trace in figure 6.18. The permanent fault case, in figure 6.21, the extinguish detecting ANN (second trace) is ignored since the algorithm correctly discerns there is no arc to detect.

Robust diagnosis: interpreting the ANN output

Clearly, a reclosure decision cannot be issued on the basis of a single window in the time domain. In the case of the arc extinguish detector, there is a transition period where the moving window takes information belonging to both domains. In one part of the window, the arc has extinguished and in the other, it is still present. It is

necessary to omit any such windows from the training set, as they do not belong to one class or the other. However, when the algorithm is online, this transition period obviously cannot be avoided since no foreknowledge exists as to when it will occur. Over this period the network exhibits confused behaviour as it dithers between arcing and extinguished (see figure 6.21, second trace). A counting mechanism is thus introduced such that a fault is only diagnosed as transient if the network response is consistently above 0.5 for one cycle. The arc is deemed to be present by the fault type detector if it lies in this domain for 2 whole cycles. This means the effect of any dither in the time domain due to unexplained phenomena is minimised. This approach ensures maximum robustness.

6-13 Conclusion

This chapter describes the algorithm development and the accompanying studies. A twin-neural network based algorithm is developed. Most importantly, the harmonic frequencies due to power electronics are specifically examined, using 3D spectrograms, and are shown not to be significant in the input to the algorithms. Provided ER 5/4 is adhered to, the algorithm should be robust to even the worst case harmonic emissions.

Chapter 7 - Real Time Deployment

7-1 Introduction

In the previous chapters 4 and 6, the RTDS-based system model is built and the adaptive autoreclosing algorithm developed for this system. This chapter concentrates on deploying the developed algorithm in real time. It presents a discussion of the hardware, the tools used in software development, the software itself, and the testing of the algorithm in a closed loop with the RTDS. This chapter also serves as a stand-alone report detailing a prototype 12-channel relay development and testing platform.

7-2 Hardware Overview

The real time development platform consists of the RTDS, the interfacing module and the pseudo relay, comprising of workstation hosting an A/D card for data acquisition. A block diagram is shown in figure 7.1.

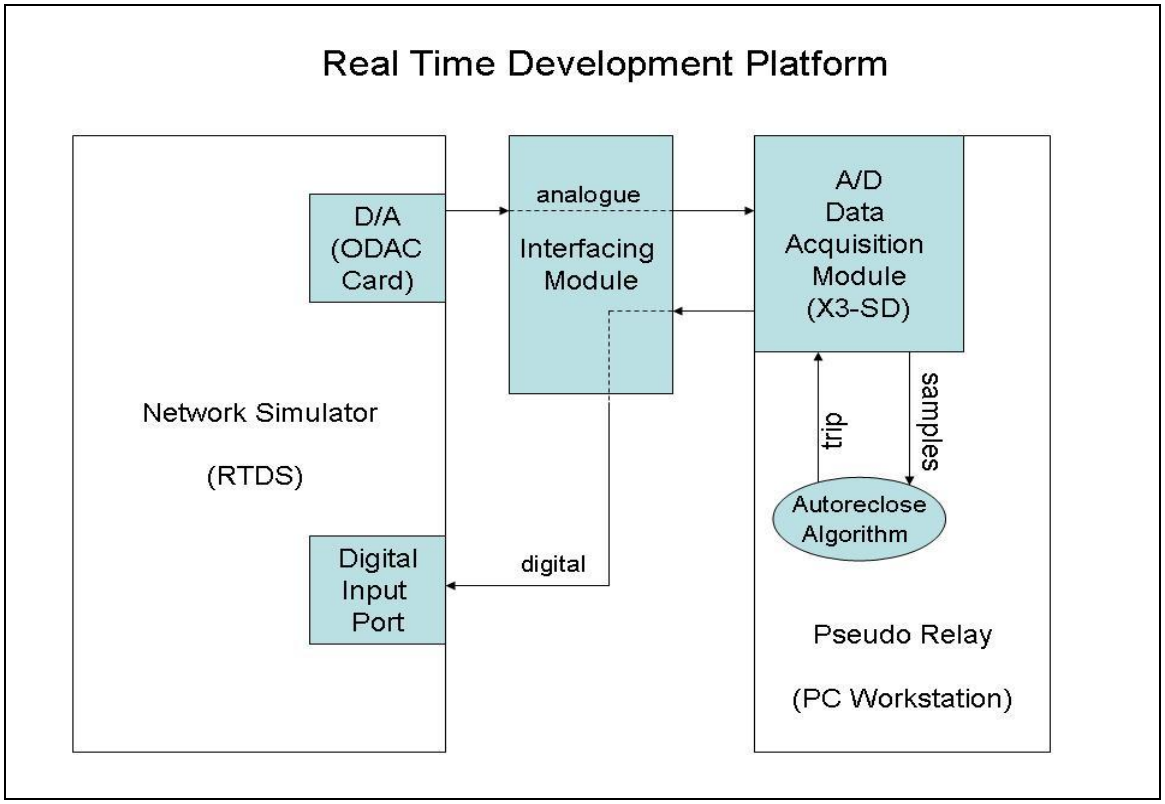


Figure 7.1: Real Time Development Platform

RTDS hardware is discussed in chapter 4. The RTDS is responsible for simulating the primary system. Analogue output is achieved through the onboard RTDS ODAC cards. Here, digital RTDS values are converted and output as analogue signals. The interfacing box is a bespoke-built enclosure for processing output signals between the two systems to desirable input levels. The data acquisition module is a high definition analogue to digital card. The product used in this work was Innovative Integration's X3-SD module. The analogue stage, although superfluous from the (digital) RTDS standpoint, is necessary so that the pseudo relay may in future be decoupled from the primary system and showcased on other platforms. It also adds an extra layer of authenticity to the pseudo relay. Data is converted to digital and streamed in packets to the workstation via the PCI express bus. The algorithm is executed in software by the PC's CPU. The desired circuit breaker status is calculated and fed back to the X3-SD. The digital signal is then processed by the interfacing module, bringing it to acceptable levels for the RTDS's digital input port.

Digital to analogue: ODAC card

Although the 3PC processor cards offer inbuilt analogue outputs, their dynamic range is not high enough for fault levels required in relay testing. Therefore, RTDS's ODAC card; a dedicated, optically isolated, digital to analogue converter is used. The 3PC cards offer two digital output ports for processors A and B, each of which are connected to an ODAC card via a shielded ribbon cable. Each ODAC is capable of three separate channels. The channels do not necessarily have to be three nodes at the same bus - the "DAC" component in RSCAD draft designates which 3PC processor will physically output any system quantities. Since Bath's RTDS system has 4 ODACs, a potential 12 Channels of analogue can be accessed from the RTDS for monitoring, protection and control purposes. More technical information about the ODAC cards can be found in [84], which is not in the public domain, but may be obtained from RTDS technologies.

Data acquisition module: X3-SD

Innovative Integration's X3-SD module is a high fidelity, analogue to digital data acquisition card with onboard 1Mgate Field Programmable Gate Array (FPGA). The module is capable of sample rates up to 216 kHz across a maximum of 16 separate channels. Oversampling and anti-alising filters are applied before the A/D

conversation process such that aliasing of frequencies above the nyquist frequency is minimised. The onboard FPGA feature is not used in this work, but be could potentially used in future for more demanding algorithms. Instead, acquired data is streamed in packets to the host PC over PCI express bus, an interface capable of data rates of up to 180MB/s. X3-SD operation is controlled via software, which can be developed using Malibu, a custom written C++ class library. An important starting point is the SNAP example software, which calibrates and controls the module in real time data acquisition.

There are no doubt numerous data acquisition and DSP development platform solutions in the marketplace. The choice of the X3-SD is fairly arbitrary, it was merely the first card to meet the required specifications. Further technical information regarding the X3-SD can be found in [85],

Host and DSP: PC workstation

The PC workstation is a standard, dual-core intel i5 650 3.20GHz machine, running Microsoft Windows XP. The workstation acts as a host for the X3-SD card, and also implements the adaptive autoreclose algorithm through the SNAP application, customised with C++ code.

Interface module

The interface module physically houses the XLR inputs from the RTDS ODAC cards, the digital out ports and the two breakout boards for the X3-SD. The digital out ports are standard 4mm test sockets. One breakout board ports the analogue into the X3-SD and a second handles the digital out. Both are connected to the module by a multi-channel ribbon cable. Figure 7.2 shows the original design for the enclosure and figure 7.3 shows annotated photos of the interface module.

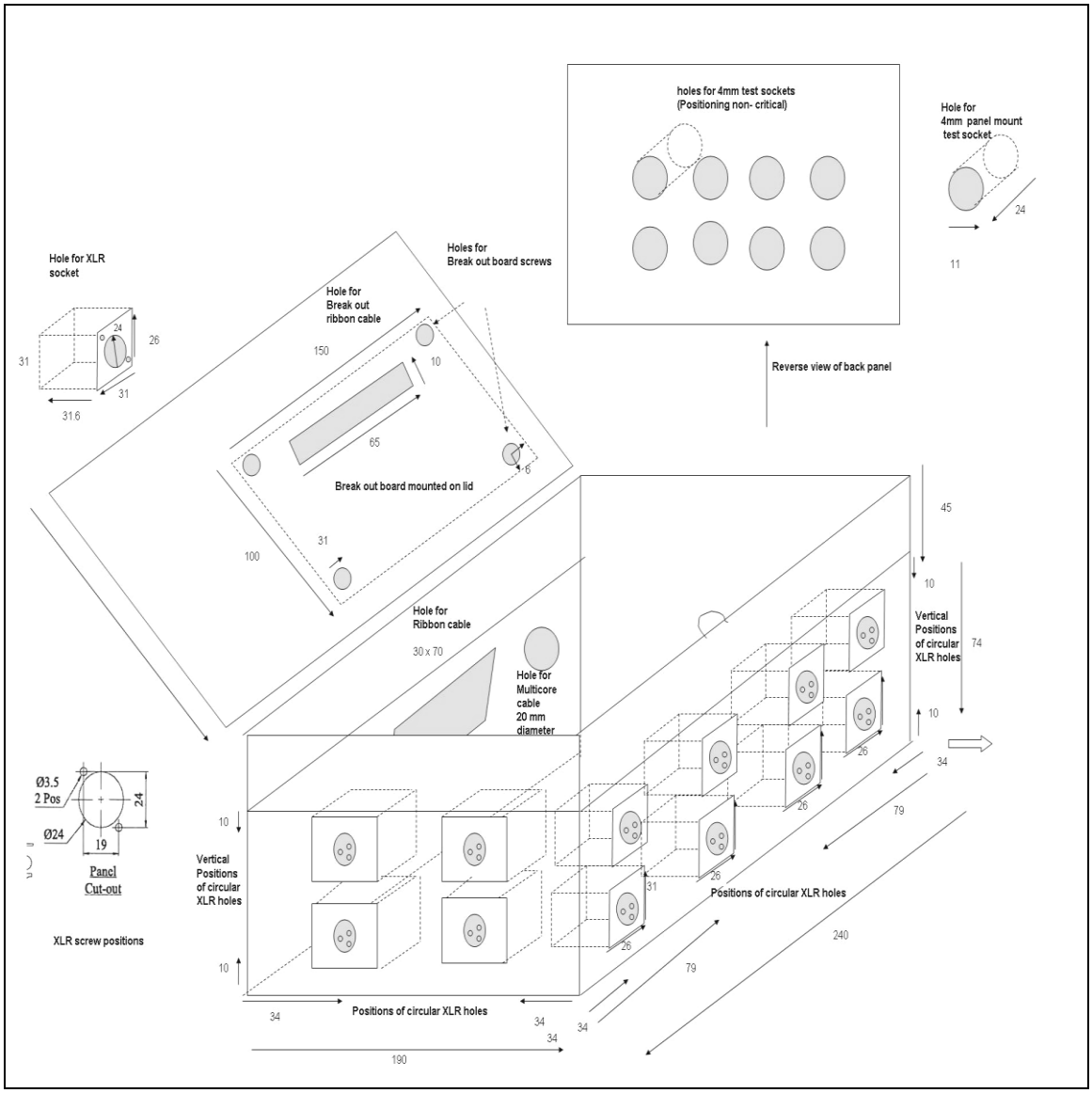
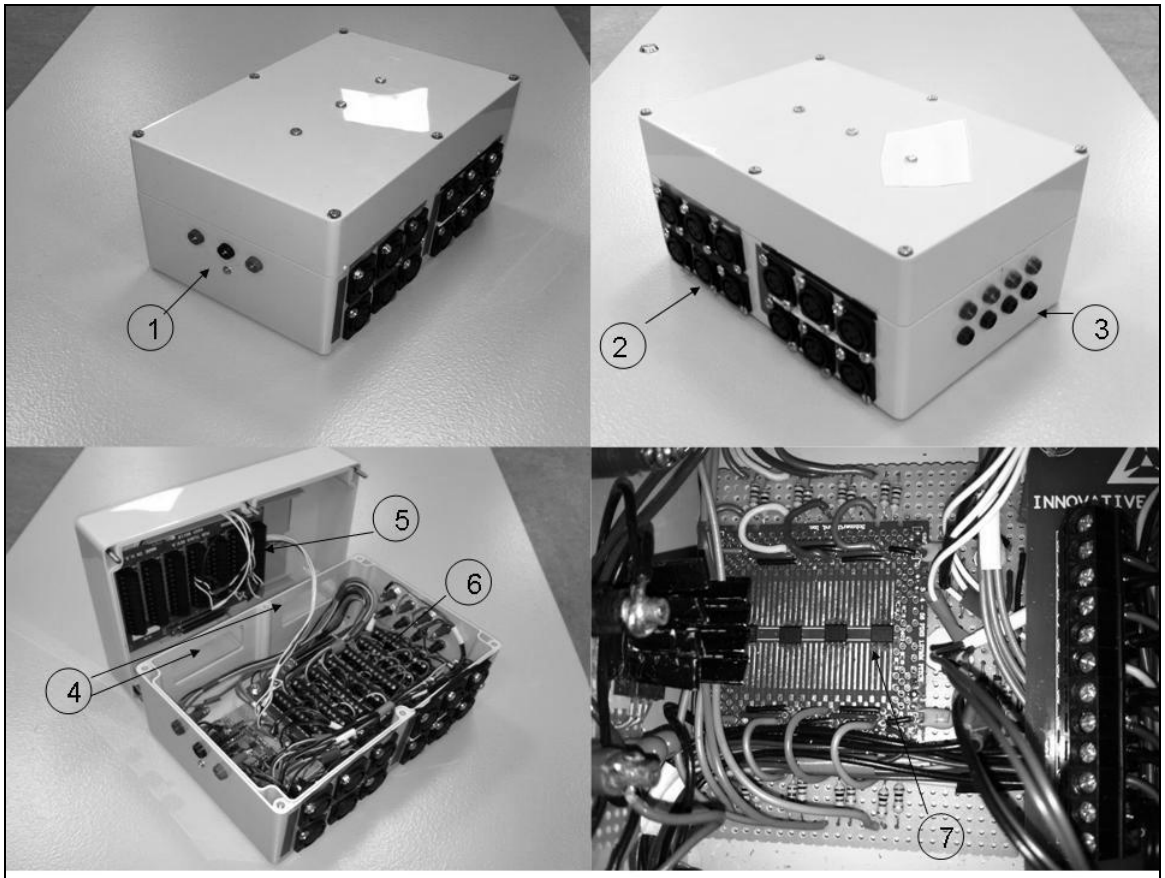


Figure 7.2: Interface module enclosure design



Key to photos

1) +/-12V DC dual power supply sockets

2) 12 XLR analogue in sockets

3) 4 channels digital out signal/common

4) Ports for ribbon cables

5) Digital out breakout board

6) Analogue in breakout board

7) Opamps in situ on IC mounts

Figure 7.3: Interface module circuitry

Interface module circuitry

The interface box is necessary to convert output signals between the RTDS and the pseudo relay to acceptable input levels. It breaks down into two areas dealing with signals to and from the RTDS.

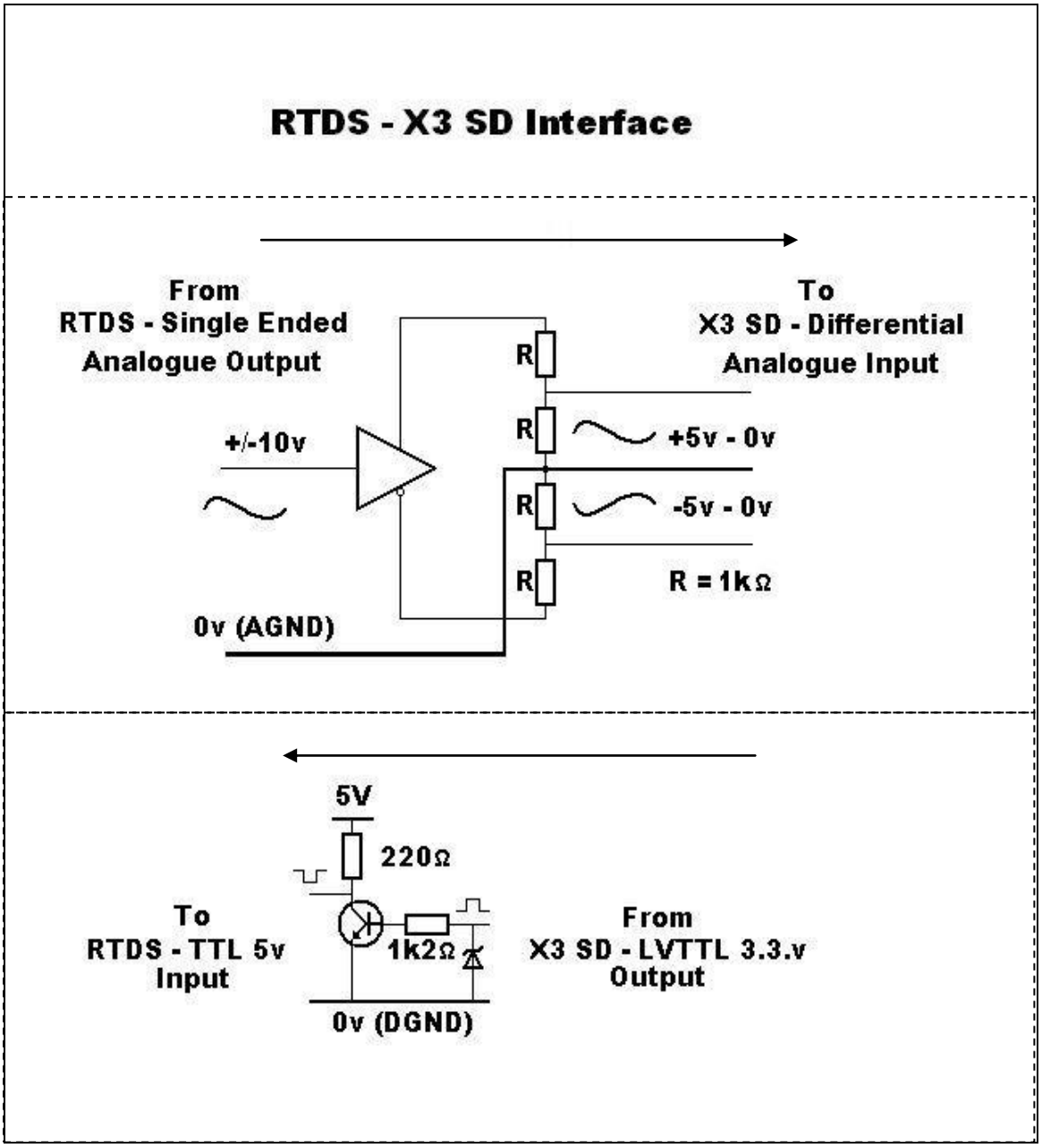


Figure 7.4: Interface module circuitry

The ODAC cards produce a single ended signal at +/-10V, whereas the X3-SD is optimised for a differential signal +/-5V to ground. The differential signal is cleaner because two sinusoids are run in anti-phase and combined to produce a third, and thus any common mode noise is rejected. It is therefore necessary to use the differential driver opamp and potential divider circuit as shown in figure 7.4. The differential driver is powered by a dual +/-12V supply rails, omitted from the diagram for clarity.

The digital out signal from the X3-SD is driven by 3.3V DC logic, whereas the RTDS digital input port takes 5V logic. The RTDS digital inputs are inverted, such that a logical 0 is indicated by a signal above the threshold. A transistor is therefore used as a switch and wired with resistors so that 3.3V into the base will result in 0V between the emitter and collector. When the base is at 0V, the collector is held at the 5V rail. The zener diode ensures that the input signal to transistor can never exceed much more than 3.3V. The 5V rail is derived, with a potential divider, from the +12V power supply rail, but is omitted from the diagram for clarity. A 12V fan was also added to the module to cool the differential drivers, this is powered from the positive rail of the supply. The opening for the ribbon cables double as cooling vents.

Psuedo Relay

The X3-SD housed in the PC workstation functions as a pseudo-relay together with its host. All the relay logic is fully customisable so any IEC 61850 compliant relay may be developed, tested, and verified using this platform. This can either be in conjunction with the interface box and the RTDS, through recorded waveforms, or a real power system. These last two options would require a different interfacing module since secondary system inputs are normally in the region of 110V.

7-3 Software

SNAP

Innovative Integration includes the SNAP program as part of their development package [86]. The software is customisable, the source and header files are included as a form based Microsoft Visual Studio 2008 project, along with support for various other development environments. The user may make changes in the code to call bespoke methods, recompiling the project to his or her requirements. The Innovative Integration ‘Malibu’ library is an extensive class library with real time optimised C++ methods, for data streaming, signal processing, DIO and other functionality that may be implemented on the host’s CPU, or the X3-SD’s FPGA. The approach taken was to use the SNAP software as a shell for the algorithm. As each packet of data is streamed from the X3-SD, individual samples are accessed and used to call the algorithm every timestep. The X3 –SD streams samples in 32-bit signed integers.

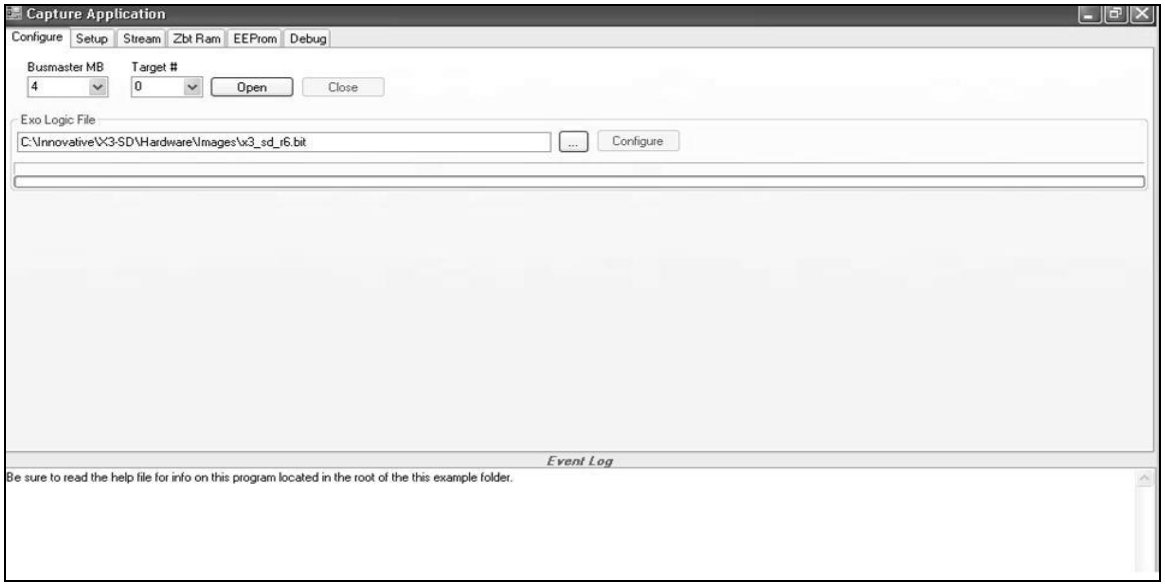


Figure 7.5: SNAP example program configuration Tab

Figure 7.5 shows SNAP's configuration Tab. Here the data acquisition module must be flashed with the firmware image file and armed for data transfer.

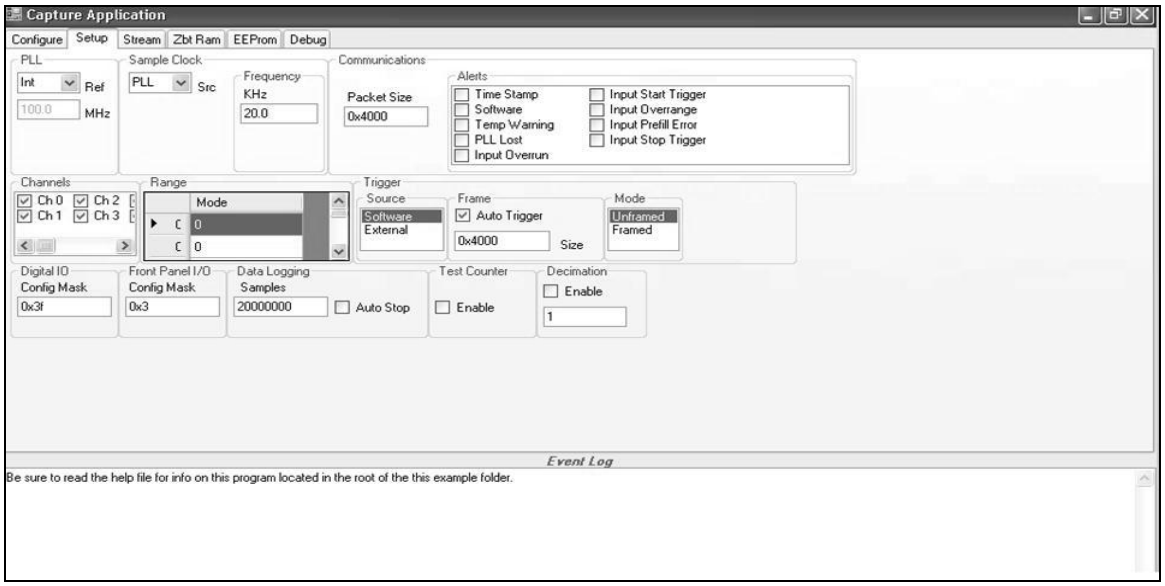


Figure 7.6: SNAP example program setup Tab

Figure 7.6 shows the setup tab. Here the sample rate, number of channels, total samples and the packet size may be selected. Various other features may be controlled but only the default settings were required for the purposes of this work.

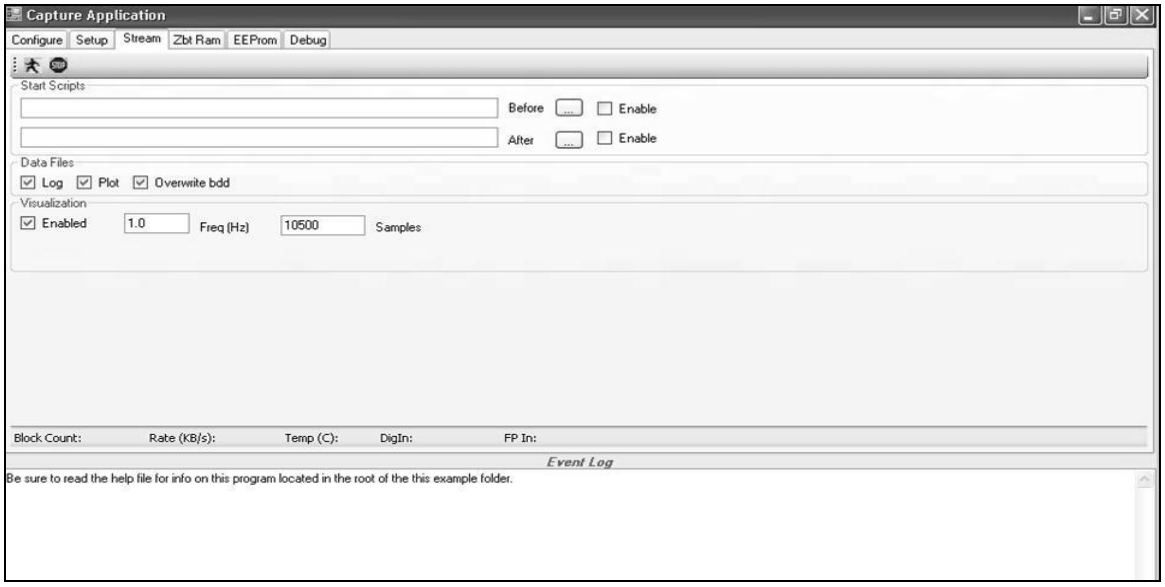


Figure 7.7: SNAP example program configuration Tab

Figure 7.7 is the stream tab. Data acquisition and streaming can be initiated from this tab along with real time visualisation via the Binview graphing utility. The software was augmented to implement the algorithm when the stream button was pressed. However, the BinView graphing utility was not selected to make sure the CPU had optimal resources to run the algorithm.

Simulink ERT

As mentioned in the previous chapter, the algorithm is developed in Matlab's simulink. An extremely useful feature of this ubiquitous software is 'Embedded Real Time encoder' (ERT), part of simulink's Real Time Workshop. This feature automatically codes the blocks into C++ optimised for real time. ERT's compiler writes fully commented code with transparent entry points. In other words, any system developed in simulink may be encoded and implemented in real time within separate hardware or software. This maybe another PC, a microcontroller, or other embedded platform. The user is not required to hand code the algorithm. The SNAP code is proprietary to Innovative Integration, and so must be omitted, but the customised code and the code produced by ERT is included in appendix 2.

7-4 Computational load assessment

An algorithm involving many Fourier transforms can be time consuming, even for modern processers. It was therefore important to establish there would be enough computing power to run FFTs in real time, with remaining overheads for the rest of the algorithm and the operating system. Floating point operations required for an N length FFT, where N is also a power of 2, are less than the expression (7.1) [83].

$$3N\log_2 N \qquad (7.1)$$

Given that the sample rate at the FFT stage is only 2.5 kHz, the algorithm must compute 2500 FFTs per second. If each FFT is of length 64 samples, this requires the load described by (7.2).

$$2500\times 3\times 64\times \log_2(64) = 2.88\times 10^6$$

$$= 2.88*10^6 \text{ FLOPS or } 2.88 \text{ MFLOPS} \qquad (7.2)$$

Where MFLOPS in (7.2) is millions of floating point operations per second. Once the ANNs are trained, their online execution depends on a small number of summing junctions and the multiplication involved in the transfer functions. The number of floating point operations *per time step* is in the order of the number of synaptic connections, plus the number of individual neurons. The fully connected MLP used here, with one hidden layer and neuronal configuration 8-12-1 would require, in the region of equation (7.3):

$$8\times 12+12\times 1=108 \qquad (7.3)$$

108 summing calculations. Each neuron, with reference to chapter 5, equation (5.3), requires a further 2 floating point operations to implement the transfer function and the bias, bringing the total to just under 150 operations. Thus, the total floating point operations *per second* for one ANN should not exceed 7.4.

$$150 \times 2500 = 3.75 \times 10^5 \qquad (7.4)$$

This must be doubled to account for the two networks used in this algorithm. The result is

$$0.75 \text{ MFLOPS} \qquad (7.5)$$

And therefore the total requirement of the algorithm is around 3.63 MFLOPS. According to the SiSoftware benchmarking software *Sandra lite* [87], a single core of the Intel I5 Processor is capable of up to 26.6 GFLOPS. A single processor is capable of up to 7,000 times the required speed. Even though the code is not parallelised to take advantage of the PC's multiple cores, computation running on a single processor should be more than adequate for the purposes of this algorithm.

This discussion assumes that considerations such as simple Boolean logic and buffering operations present negligible load on the CPU. It also assumed that the FFT and ANNs are programmed with lowest level routines, i.e. with maximum theoretical efficiency and require no unnecessary calculations. Also, external to the algorithm, other real time procedures in the SNAP software are assumed to present negligible load, as well as overheads for the background operating system. These include writing packets to memory as they are streamed from the X3-SD module. In reality, these may be far greater than the algorithm itself. Given the X3-SD is engineered and optimised for far more demanding applications, these operations should be well within its capabilities. It is also worth mentioning that DSP-optimised or even FPGA implementation on a prototype device will be much faster since the hardware and software are optimised for real time, and the operating system is skeletal at most. Thus, the real time discussion presents a worst case scenario: if real time is possible using the pseudo-relay workstation, it will certainly be possible in a dedicated modern IED.

7-5 Algorithm Testing

Data visualisation is available from the snap program via the BinView utility shown in figure 7.8. This program graphs the raw binary data that is received and streamed

by the X3-SD. The two signals on figure 7.8 indicate the faulted phase and the circuit breaker status. The circuit breaker indicates when to activate the algorithm and is derived from a second RTDS output channel. Information before this time is not relevant as no secondary arc is present.

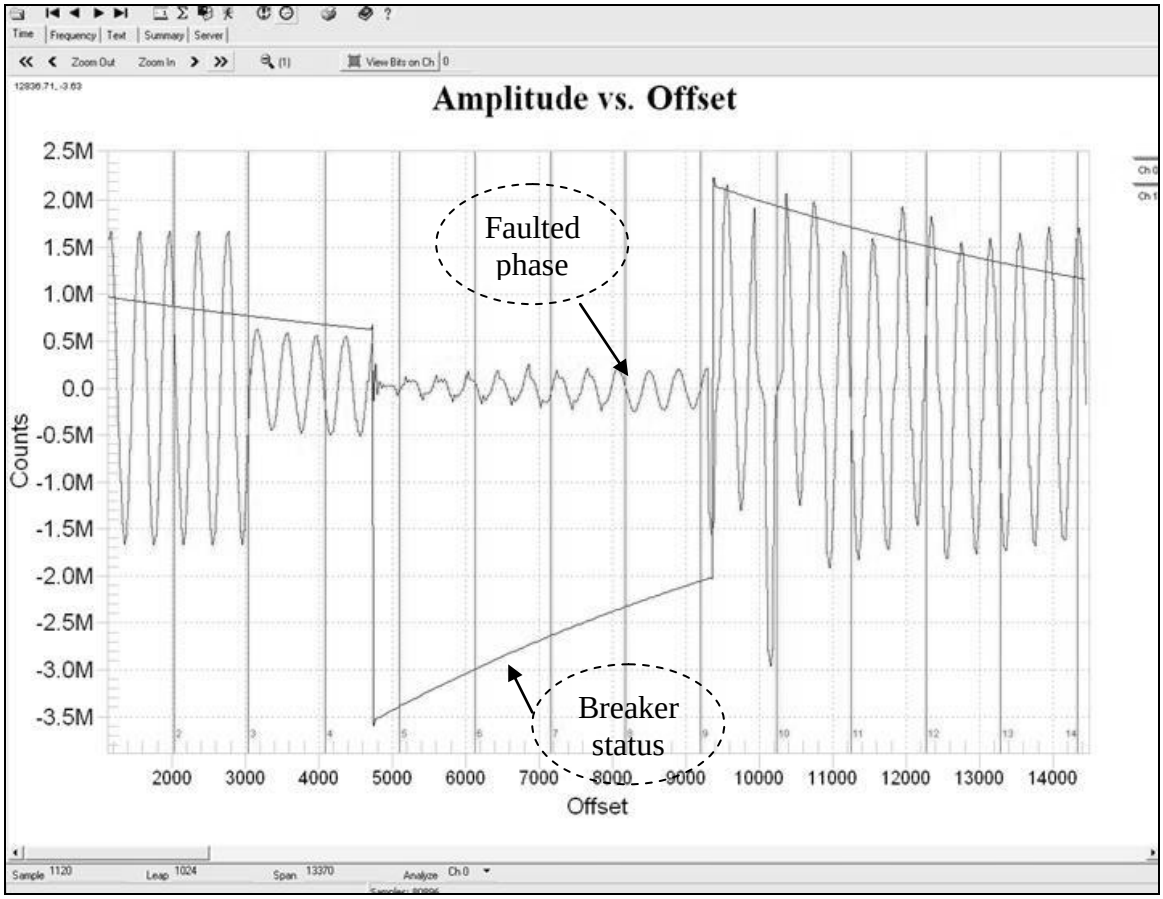


Figure 7.8: SNAP example program configuration Tab

Data visualisation is also available via the runtime component of the RSCAD software suite. The Runtime program also affords monitoring of other areas of the power system, other than the relay's input from the CVT, so results are presented using this graphing utility. However, this does not give insight into the response of the ANNs. Offline examples are shown through simulink's scope outputs in the previous chapter, in figures 6.18 and 6.19.

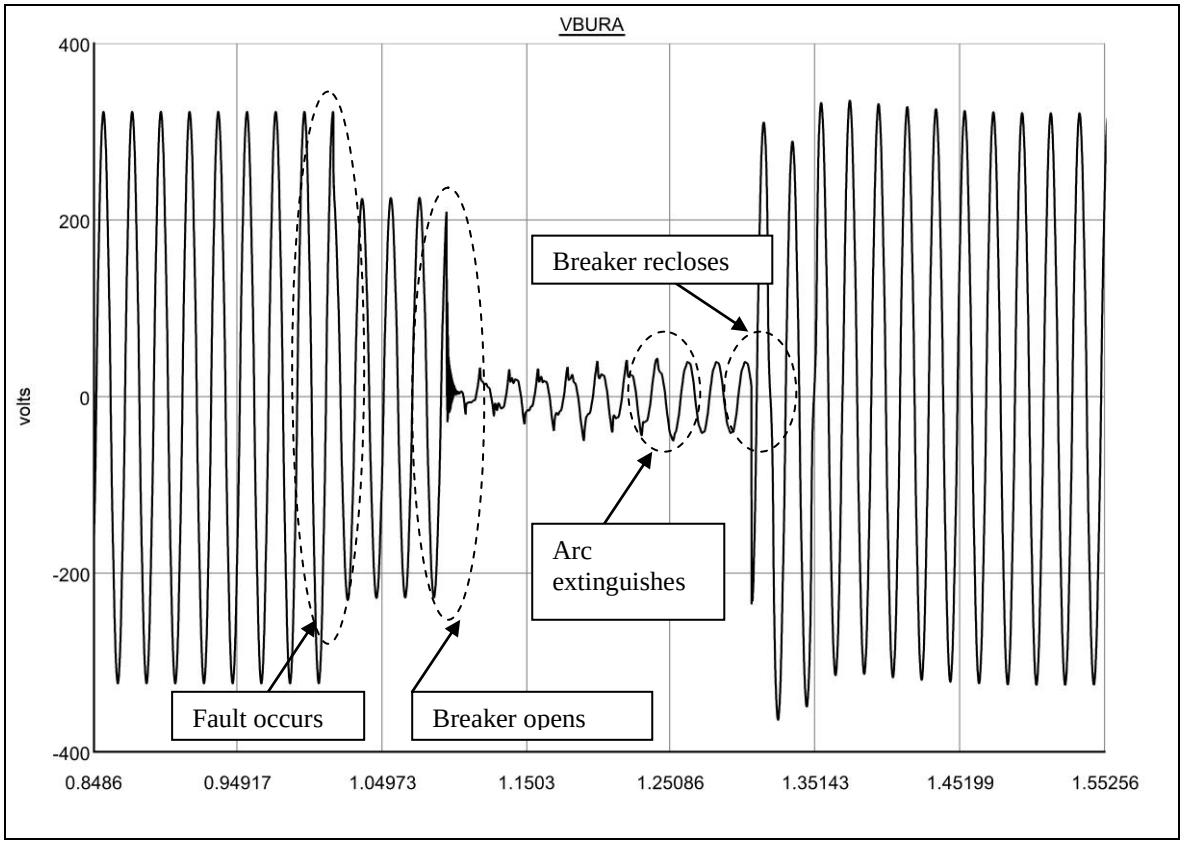


Figure 7.9: Real time response of CVT to transient fault

Figures 7.8 and 7.9 show a typical single phase autoreclose sequence as monitored by the algorithm. This is the CVT response, so it is important to verify that the power system is responding in the same way. The corresponding bus values at Farr are shown in figure 7.10.

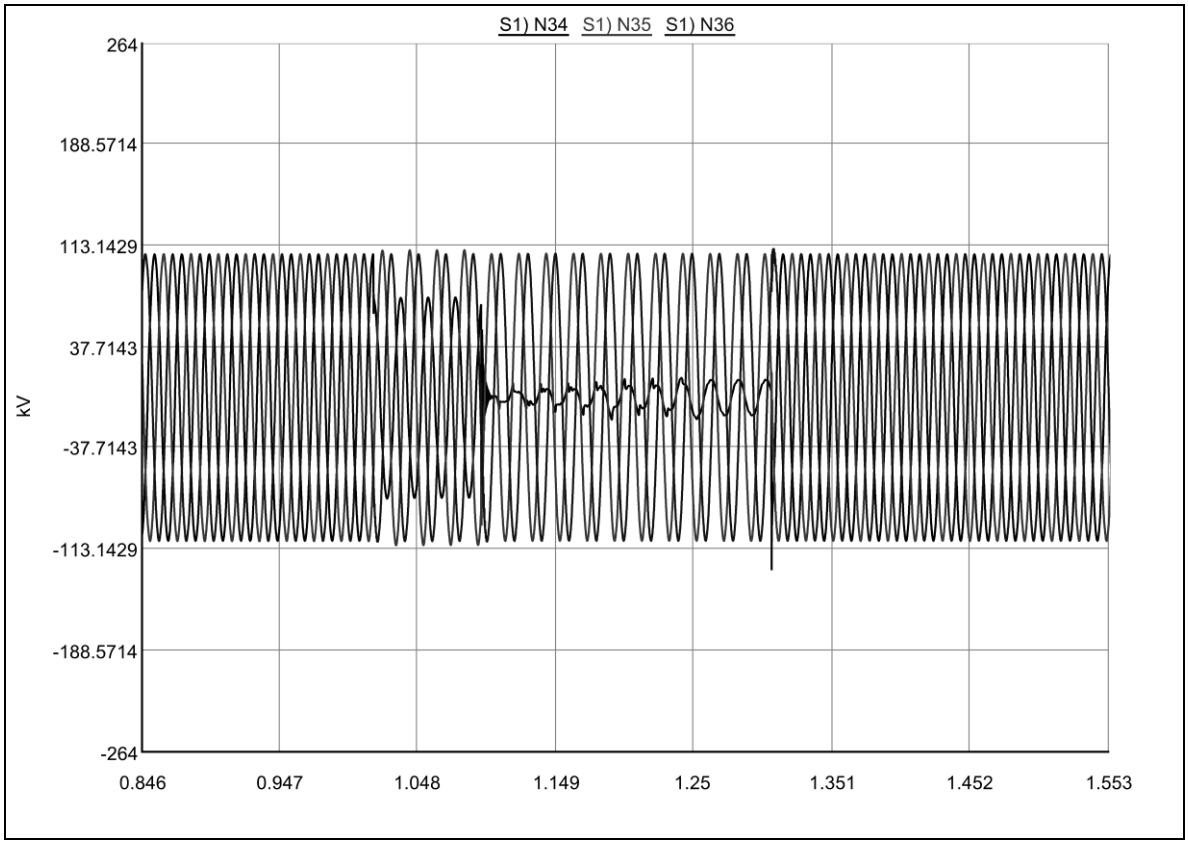


Figure 7.10: Farr bus response to transient fault

The faulted phase response is almost identical, albeit on the CVT trace, figure 7.9, there is a slight DC offset in the first cycle after reclosure.

Figure 7.11 show the phase currents measured through the line section nearest the relay bus. They show a large p.u. short circuit current before circuit breaking. The current falls to zero whilst the phase is open circuit, and following autoreclosing the small steady state current is re-established.

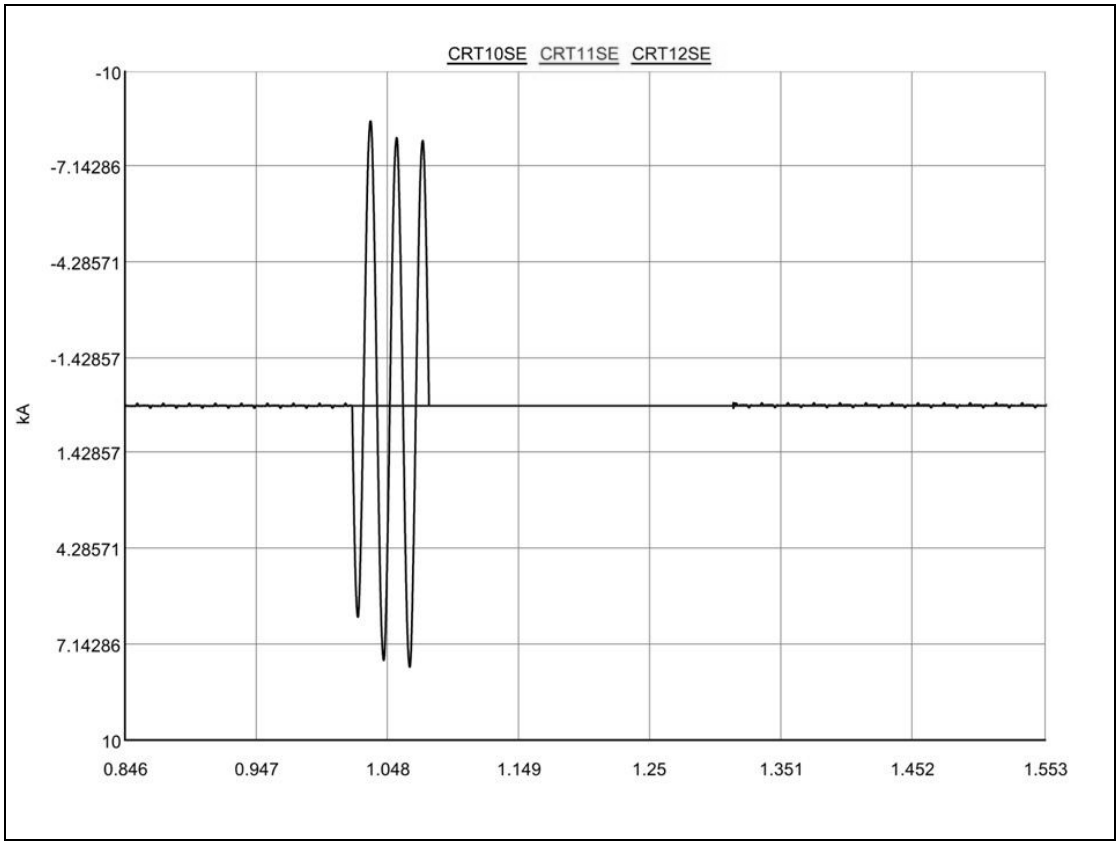


Figure 7.11: Current in Farr line sections, transient fault

The typical response to a permanent fault is shown in figure 7.12. As intended, autoreclosure is not authorised.

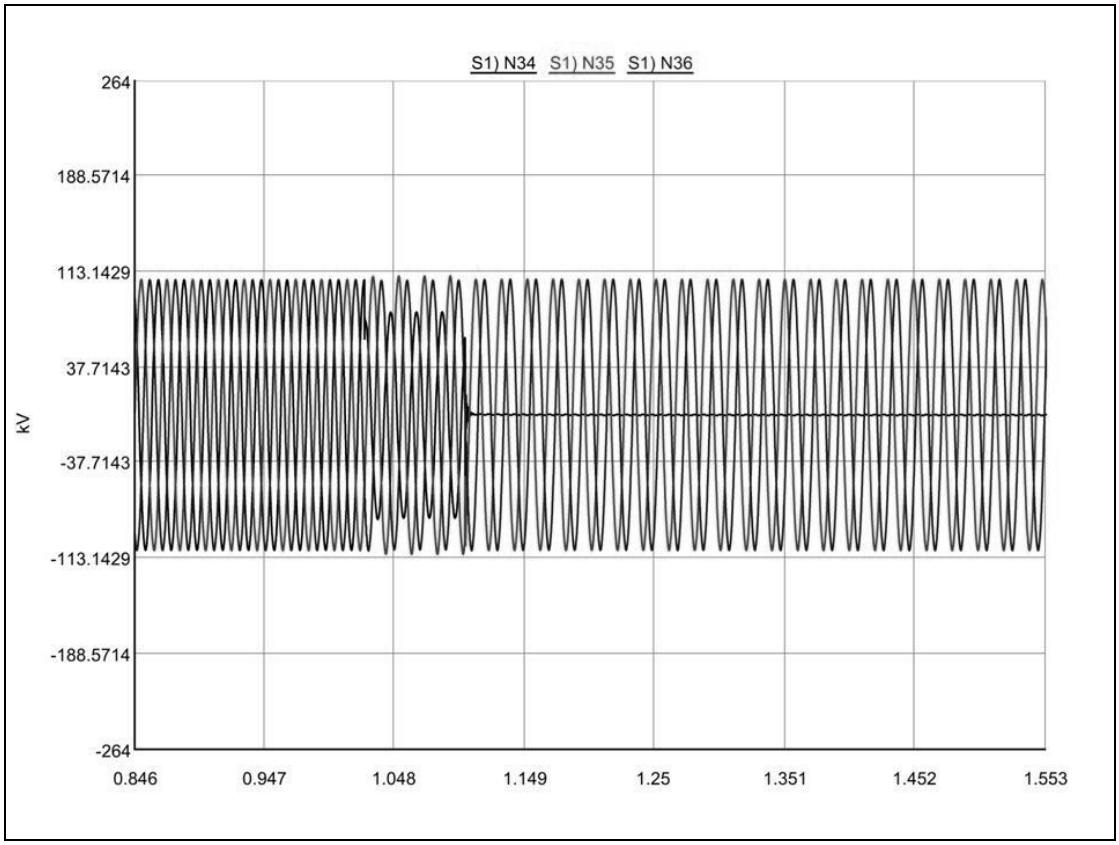


Figure 7.12 Farr bus response to permanent fault

The corresponding CVT response is shown in figure 7.13. There is a short DC offset due to the residual charge and/or remnant flux in the CVT after the circuit breaker. The algorithm is robust to this effect for two reasons: firstly, it ignores breaker response in the initial cycle following breaker opening, and any remaining variation in DC-offset is dealt with in ANN training.

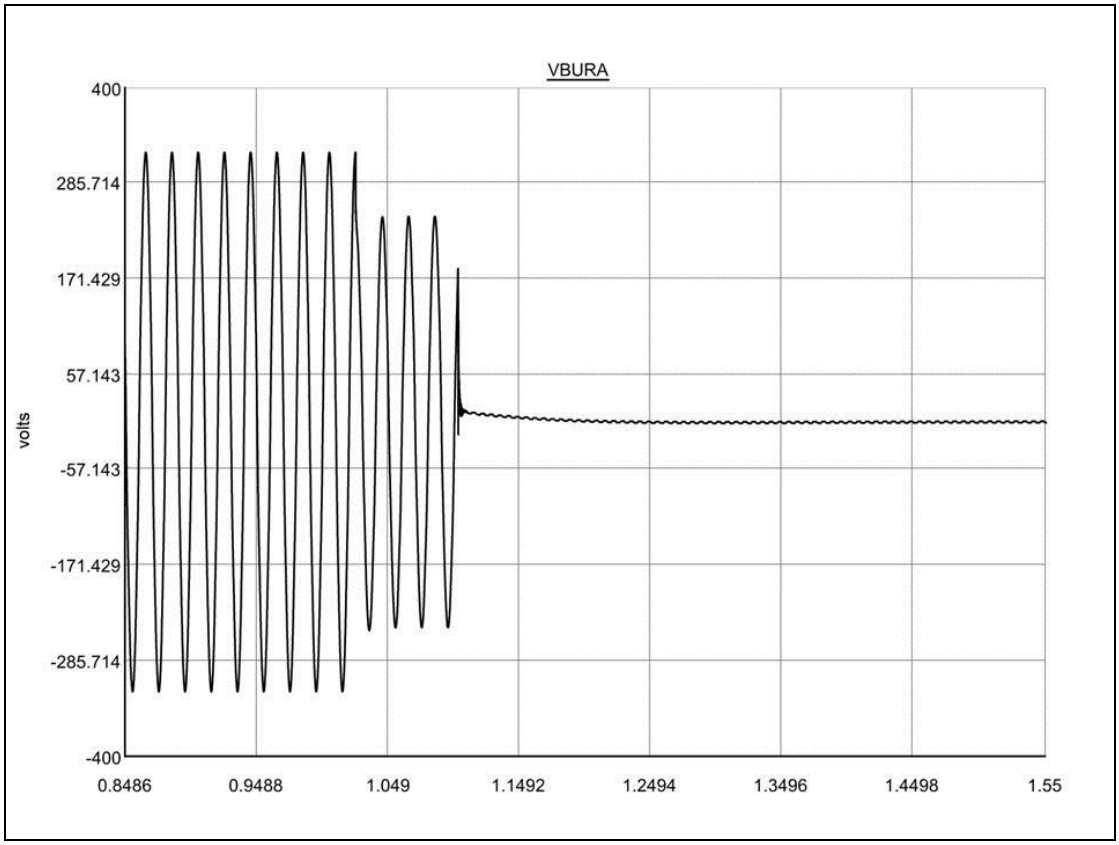


Figure 7.13: CVT response to permanent fault

Forced Mal-operation

What are the consequences of a wrong decision? If the autoreclose signal is forced to be non-adaptive and the algorithm is bypassed, the behaviour is shown in figure 7.14 (reclosing onto permanent fault) and figure 7.15 (reclosing onto transient fault before arc extinguish). In the permanent fault case, figure 7.14, the effect is a permanent resistance causing the same voltage collapse as is seen post fault, pre-circuit breaker operation. In the case of a transient fault, figure 7.15, the reclose is forced before arcing has extinguished, and primary arcing begins again. The voltage traces after the forced, erroneous reclose are similar because the primary arc resistance is similar to the permanent resistance. The adverse consequences to the system are best illustrated through the recurring short circuit line currents, shown in figure 7.16. This is shown for the permanent fault, but the transient case is almost identical. In practice, protection would trip a second time, but only after a second burst of large short circuit current. The damage sustained by the system is likely to be at least doubled since ohmic heat cannot dissipate in this short interval. Thus, fast single phase autoreclosing must be intelligent and guaranteed successful if it is to be employed in this context.

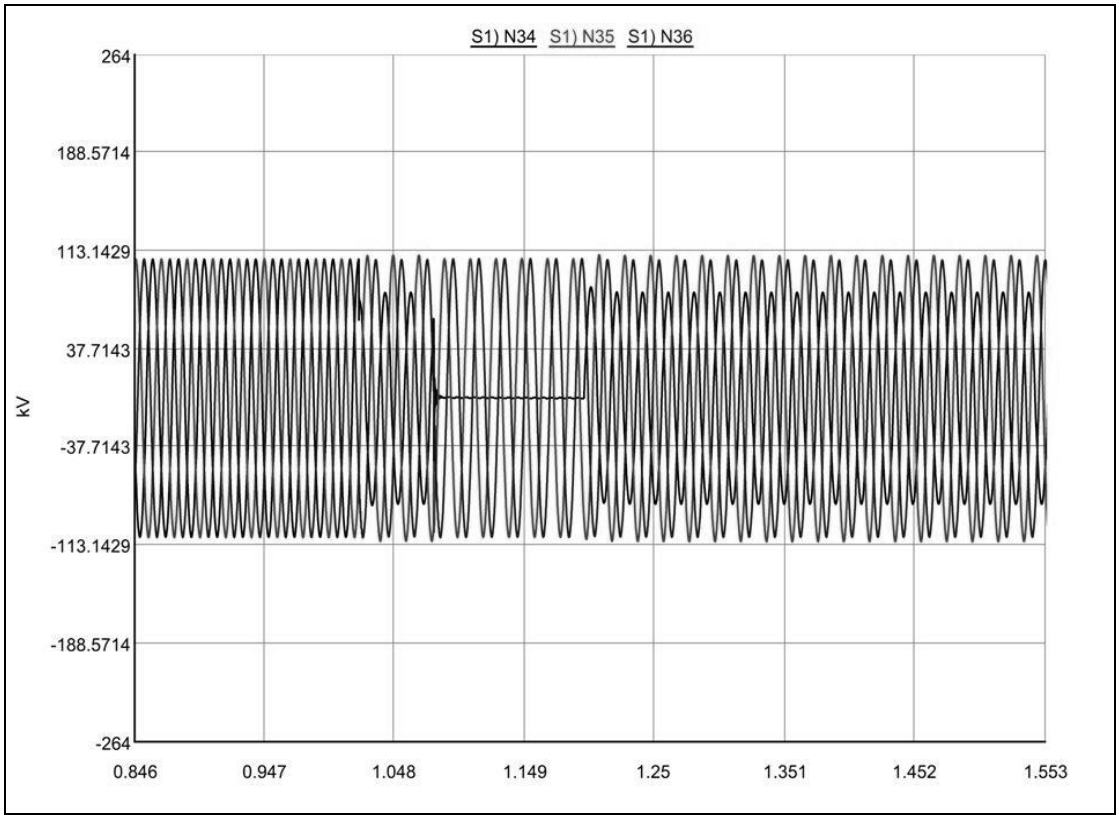


Figure 7.14: A failed reclose onto permanent fault, Farr bus

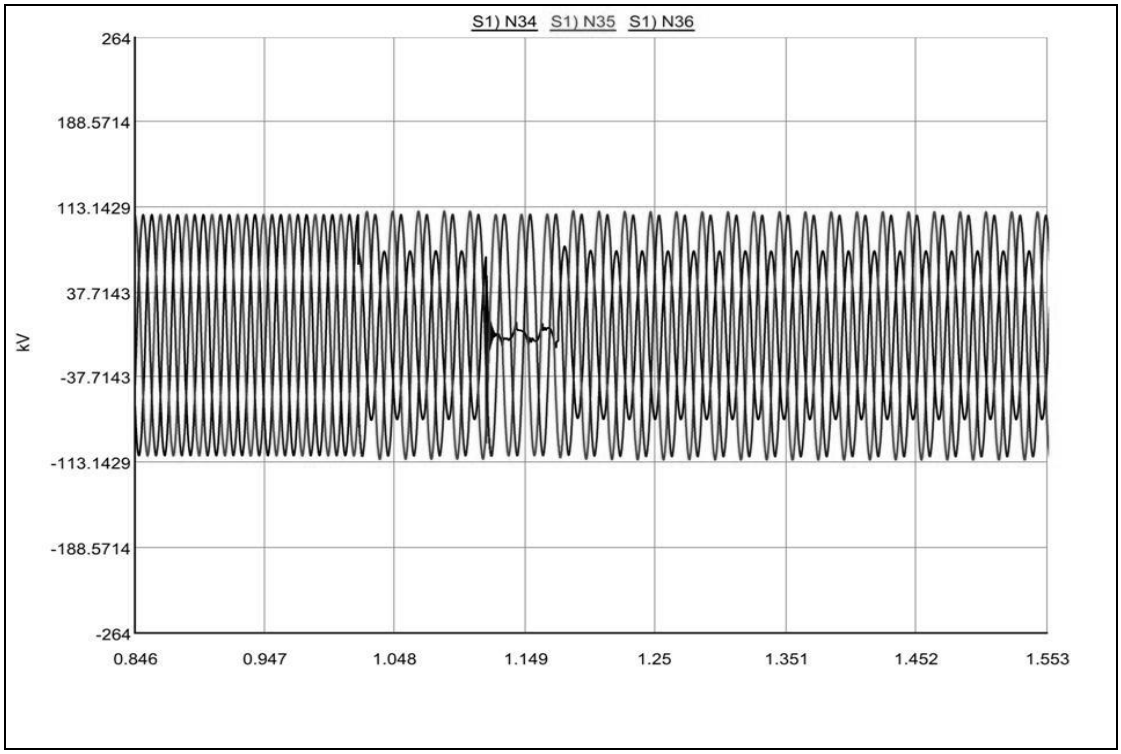


Figure 7.15: A failed reclose onto transient fault before arc extinguish, Farr bus

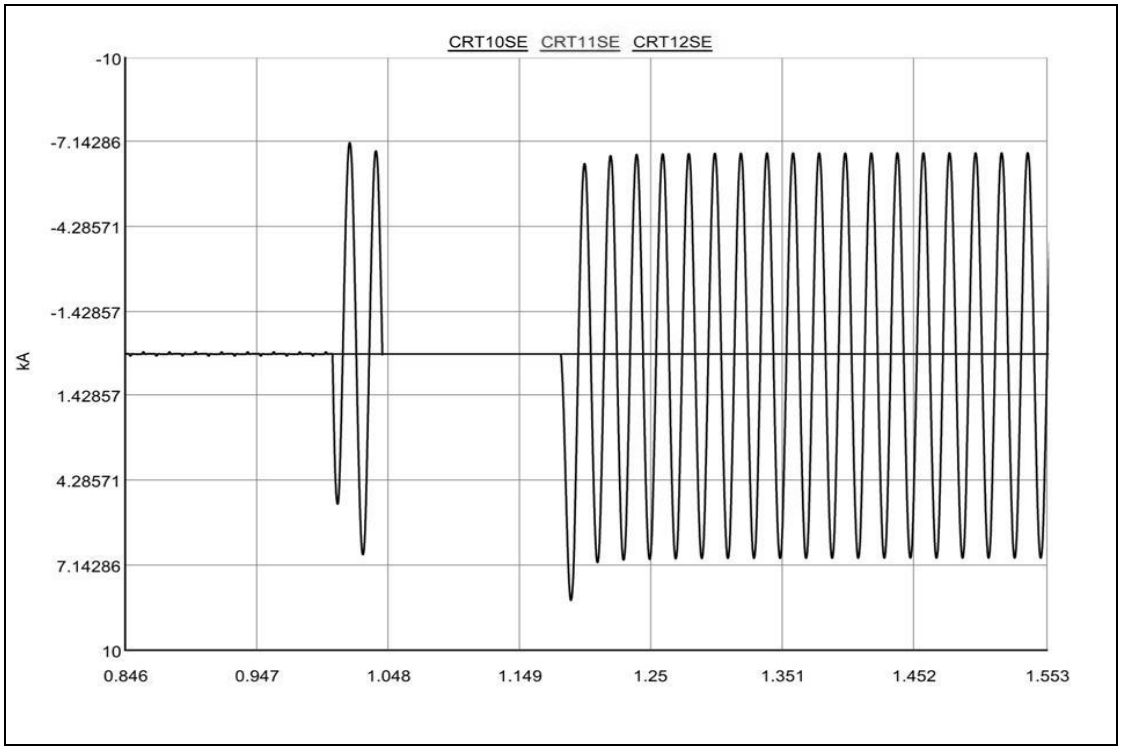


Figure 7.16: Line current response to permanent fault

Algorithm Testing

To ensure the ANNs are able to generalise rather than just memorise, it is necessary to test the algorithm with cases that are not used in the training data. Specifically, this means testing with the presence of wind farms in the system, and varying the fault point and fault inception point on the waveform. In some cases, the line length was below 15 km, and since this is below the travel time for the RTDS 50μs time-step, a simple PI-section lumped parameter model was the only option. However, this approximation is less onerous for smaller line lengths [88], and mixing transmission line models in training and testing stages had no observable effect on the performance of the algorithm, further testament to its robustness. For brevity, the results of all cases are presented in tabular format, in Table 7.1.

Table 7.1: Real time testing of algorithm

FAULT TYPE	FAULT LOCATION FROM RELAY	FAULT INCEPTION POINT	AUTO-RECLOSE	CYCLES AFTER EXTINGUISH	SYSTEM RECOVER	ALGORITHM SUCCESSFUL
T	16.60%	MAX	N	No Extinguish in simulation	N	**Y**
P	16.60%	MAX	N	N/A	N	**Y**
T	16.60%	MID	N	No Extinguish in simulation	N	**Y**
P	16.60%	MID	N	N/A	N	**Y**
T	16.60%	MIN	N	No Extinguish in simulation	N	**Y**
P	16.60%	MIN	N	N/A	N	**Y**
T	33%	MAX	Y	2.5	Y	**Y**
P	33%	MAX	N	N/A	N	**Y**
T	33%	MID	Y	2.5	Y	**Y**
P	33%	MID	N	N/A	N	**Y**
T	33%	MIN	Y	3.5	Y	**Y**
P	33%	MIN	N	N/A	N	**Y**
T	66%	MAX	Y	3	Y	**Y**
P	66%	MAX	N	N/A	N	**Y**
T	66%	MID	Y	3	Y	**Y**
P	66%	MID	N	N/A	N	**Y**
T	66%	MIN	N	2	N	**Y**
P	66%	MIN	N	N/A	N	**Y**
T	83.40%	MAX	Y	3.5	Y	**Y**
P	83.40%	MAX	N	N/A	N	**Y**
T	83.40%	MID	Y	3	Y	**Y**
P	83.40%	MID	N	N/A	N	**Y**
T	83.40%	MIN	Y	3	Y	**Y**
P	83.40%	MIN	N	N/A	N	**Y**

Table 7.1 shows that the algorithm behaved correctly in all cases it encountered. The varying real time response of the reclosure signal is down to the discrete packet

stream in the SNAP example software. Each incoming packet of samples is sent to the algorithm for processing. If a reclosure is authorised this is only sent to the RTDS at the end of the packet. The packet size was set to 400 samples, 200 per channel, see figure 7.6. At the initial sampling rate of 20 kHz, this represents 200 per cycle. This represents one packet per cycle, so the autoreclose may only be in discrete steps of one cycle. Appendix 2 gives more details of this, particularly the for loop in the 'handledataavailable()' method, in the source file 'ApplicationIO.cpp'.

An interesting case in the testing was when the fault was located at 16.6% (5 km on 30.1 km line). The secondary arc did not extinguish in this case, as is shown in figure 7.17. This would need to be initially treated as a permanent fault, but may be cleared by tripping all three phases and then attempting reclosure. The algorithm may be easily extended to deal with this by looking for arc extinction after three phase breaking and reclosing. Moreover, it is worth noting the algorithm performed correctly within the scope of its design. In this instance, the fault type was diagnosed to be arcing, but autoreclosure was not authorised since arc extinction was not detected.

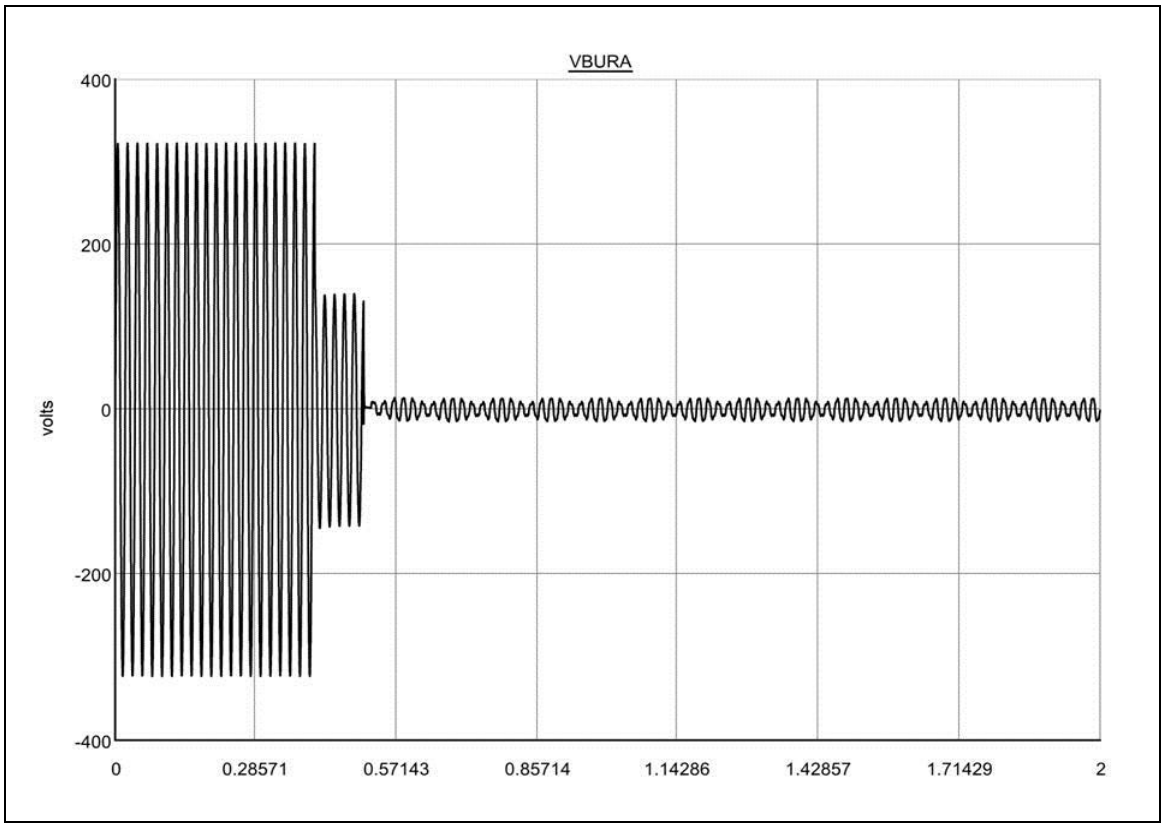


Figure 7.17: CVT response to transient fault, no arc extinguish

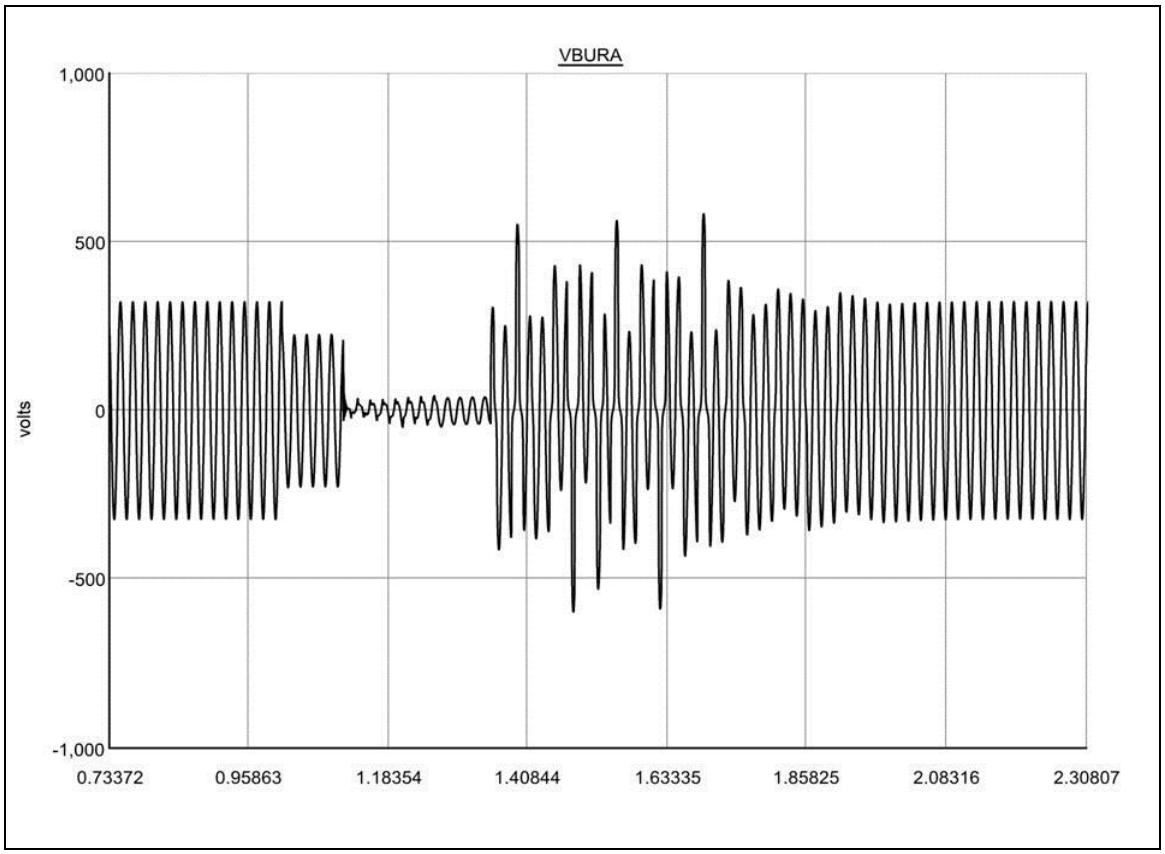


Figure 7.18: Ferro-resonance in CVT

In some cases, the CVT response was considerably different from the power system response at the associated phase. Occasionally, a ferro-resonance condition was excited by the autoreclosure event. An example of this is shown in figure 7.18. This phenomenon occurs at low burden values on the CVT, and thus is particularly relevant to reclosing schemes. The condition is likely to be excited when the flux in the core of the CVT is at a maximum [88] and the voltage is at a minimum (low burden). Clearly, CVT ferro-resonance must be avoided or other protection and control equipment will mal-operate. A CVT includes a ferro-resonance filter that is carefully designed to suppress this condition. With reference to chapter 4, section 4-4, the CVT used in the RTDS simulation was assumed to be nominally the same design as a 230kV CVT, with values scaled for a different power system voltage and frequency. The ferro-resonance filters on the 132kV CVT may therefore be designed differently or values supplied in [59] be inaccurate. More work concerning CVT modelling is necessary to determine whether the adaptive autoreclosing scheme must

be augmented to take ferro-resonance into account, or merely the filter in the simulation is inaccurate.

Three phase autoreclosing and inter-circuit coupling

Since almost all circuit breakers in the UK are 3 phase, it would be beneficial to design an autoreclosing algorithm that operates within such an environment. It was shown in the previous chapter that primary arcing, before circuit breaking, cannot be detected using these methods. However, the secondary arcing due to inter-circuit coupling is possible to observe and detect and can be seen in figure 7.19.

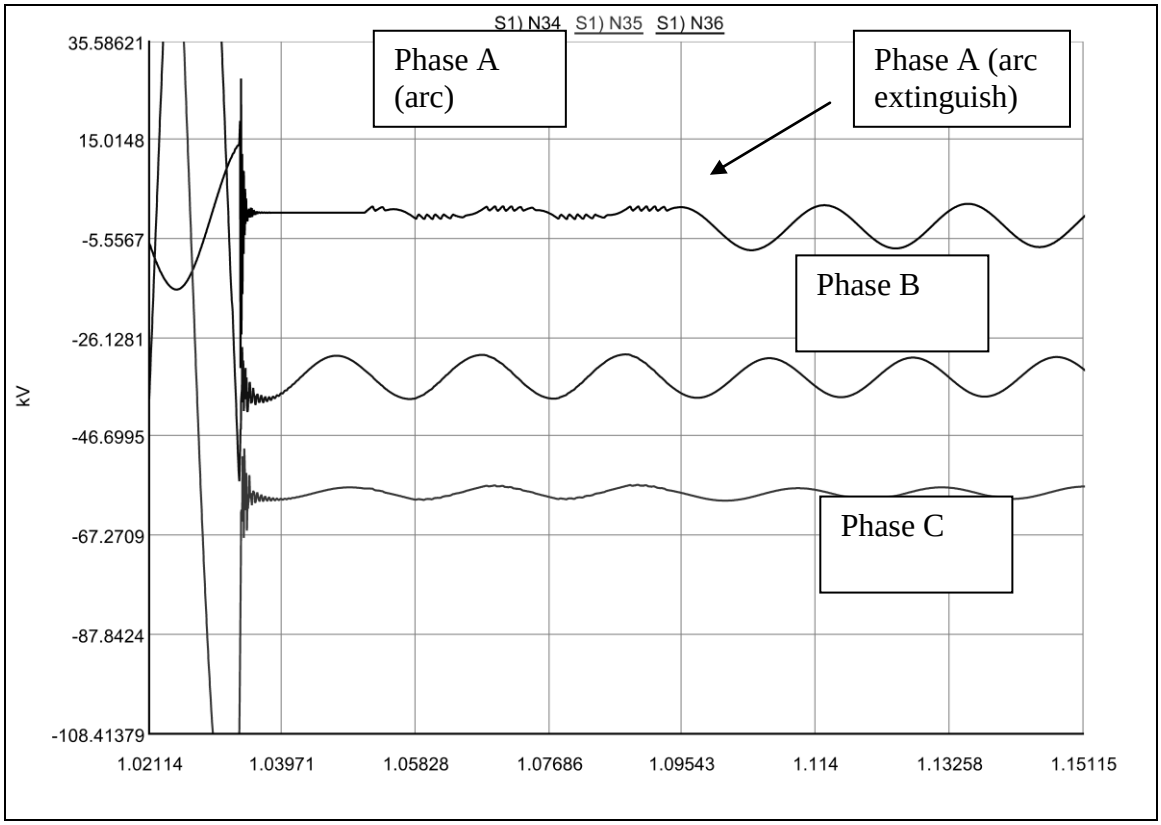


Figure 7.19: Secondary arcing on A phase and mutual inter-circuit *coupling on other phases. This arc is atypically long for this system.*

However, figure 7.19 is atypical since it shows the longest inter-circuit secondary arc observed. Generally, for this line configuration, the secondary arc tends to extinguish extremely quickly, often in less than one cycle. Figure 7.20 shows a more typical inter-circuit arc fault, and again, close up in figure 7.21.

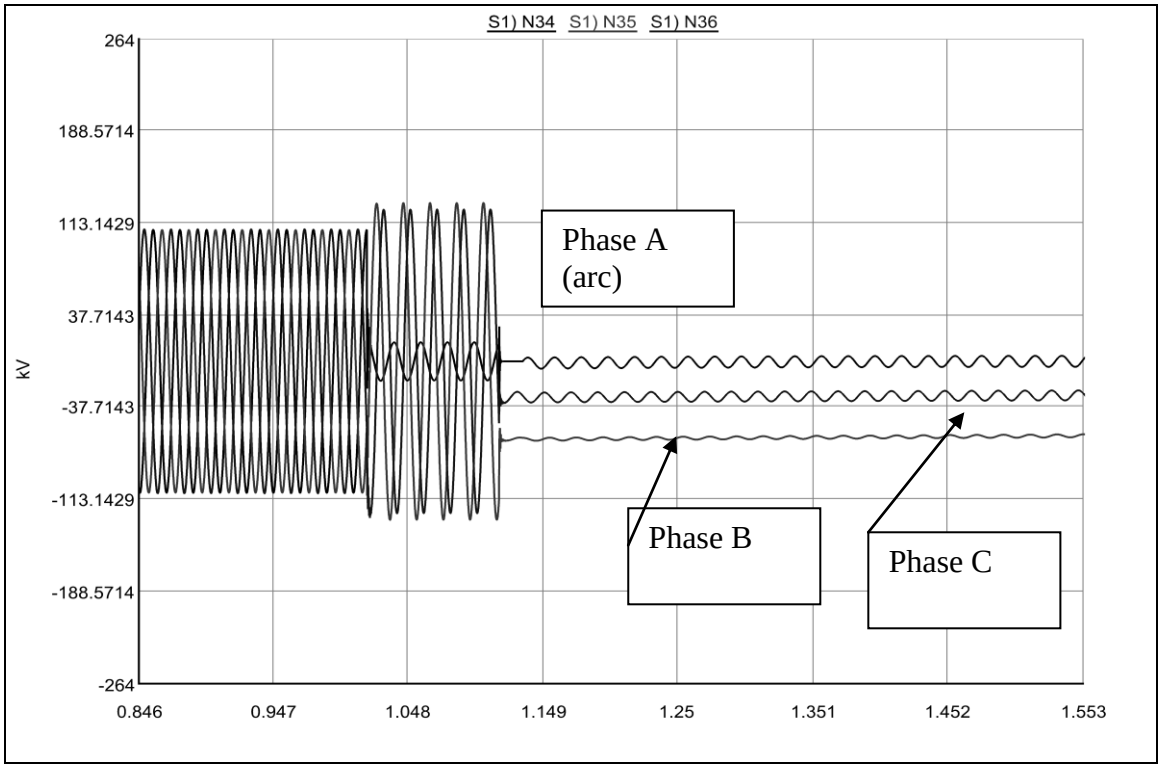


Figure 7.20: Typical secondary arcing on A phase and due to inter-circuit *coupling*

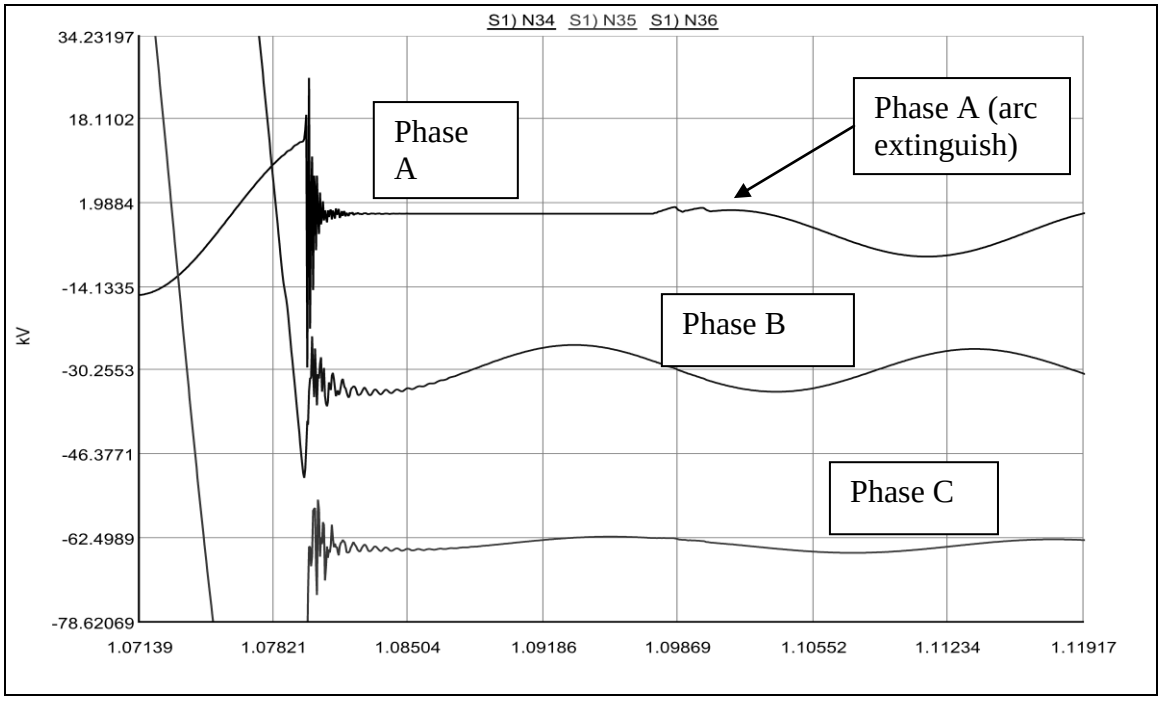


Figure 7.21: (Figure 7.20 close up), Typical Secondary arcing on A phase and due to inter-circuit *coupling*

The algorithm cannot rely on arc detection in this period because of the transients caused by the circuit breaker operation. This anomalously long arc in figure 7.19 can

most likely be attributed to the stochastic nature of the arc model as it relies on random number generators. The location of the fault on the line will also have some influence on arc duration. In untransposed lines such as this, the geometric positioning of the phase has significant bearing on secondary arc duration. There was however, a notable post arc difference between the two fault types observed at the relay bus. Following the post-breaker transients, the permanent fault had much smaller power frequency magnitude. Figures 7.22 and 7.23 show this and should be compared directly with 7.20 and 7.21. This is because the faulted phase is grounded through the fault rather than completely isolated, as it is in the transient case. This causes the observed voltage collapse.

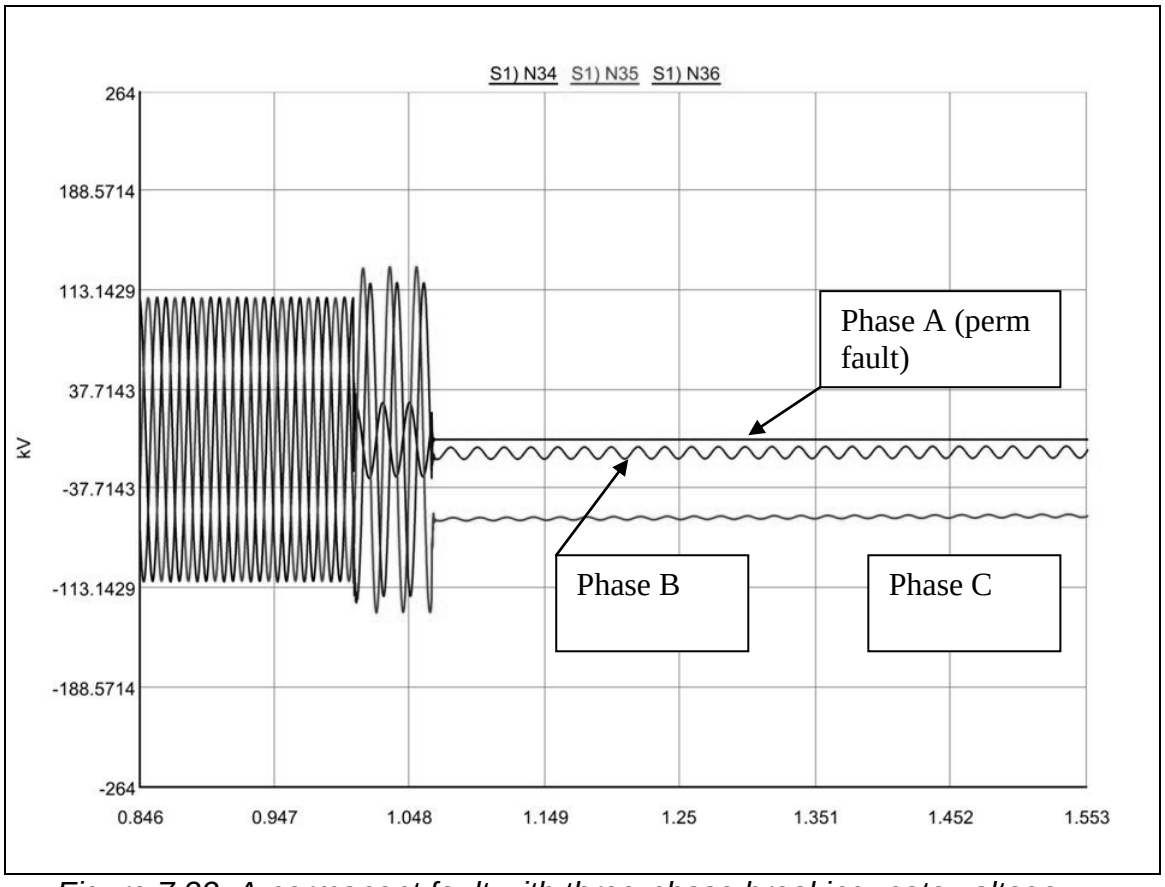


Figure 7.22: A permanent fault with three-phase breaking, note voltage collapse on faulted phase

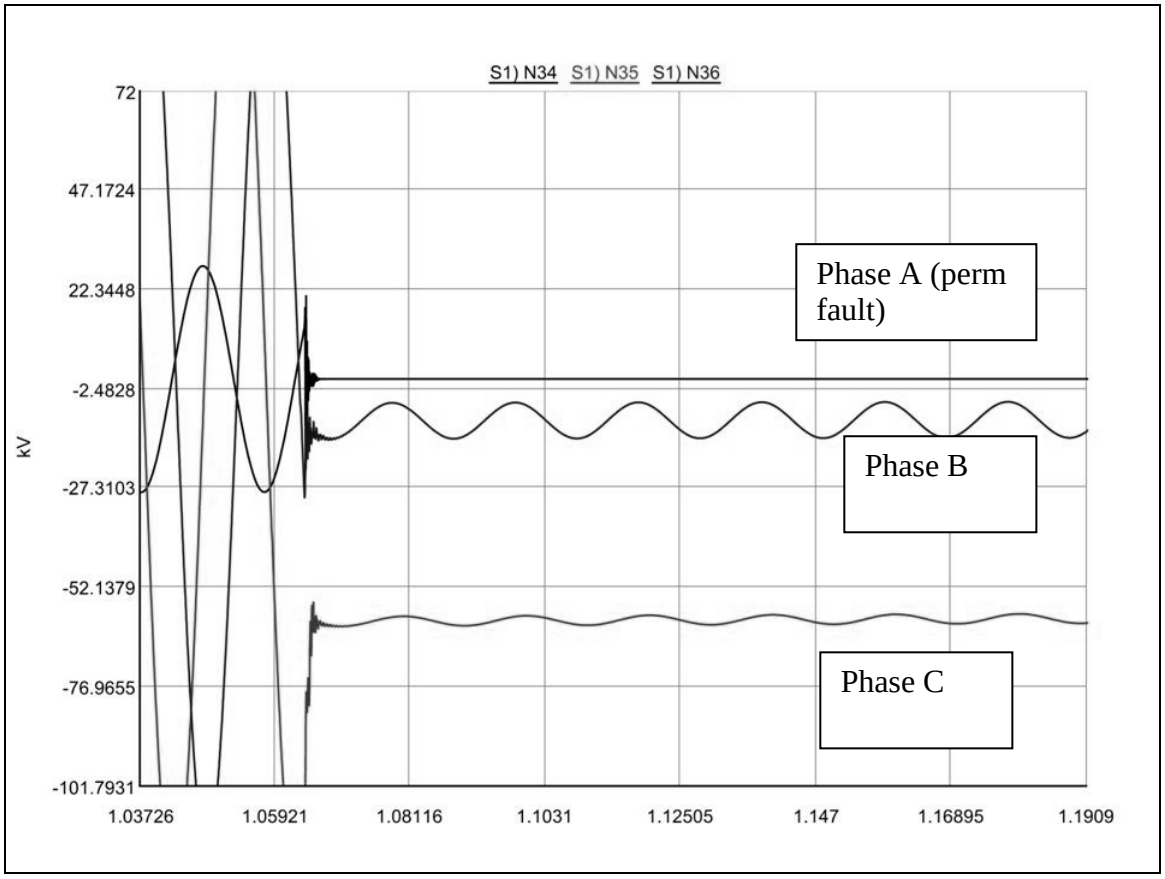


Figure 7.23: (Figure 7.22 close up), Permanent fault with three-phase breaking, note voltage collapse on faulted phase

The developed algorithm could be readily adapted to detect the fault type with appropriate ANN training. However this would be less robust as such an algorithm would rely almost exclusively on the 50 Hz component - the high frequency information caused by the arc is too brief to utilise. This does have the advantage that it would certainly not be affected by wind turbine power quality issues. Such an approach clearly relies on the neighbouring circuit being live to drive the secondary arc, and assumes that the other circuit does not experience a simultaneous single phase to ground fault from the same cause, e.g. the lightning strike. However, there are other important aspects to inter-circuit faults that determine the post breaker response, for example, conductor transposition and line configuration. Autoreclosing using inter-circuit coupling is therefore a suggested further area for research.

Algorithm sensitivity to system voltage

Some time after these experiments were completed, the algorithm was tested on a different RTDS primary system - specifically on a 50 km, 400kV line. This was for

the purposes of showcase and validation within the FlexNet research consortium, the sponsor of this work - discussed further in the concluding chapter. Without wishing to interrupt the flow of the thesis, the author believes it is important to include a short account of these tests, as they increase the usefulness of the algorithm described in chapter 6. The major difference in the 400kV system, along with the nominal voltage, is the distance between the arcing horns, which was taken to be 3.5 m rather than 0.5 m, contributing to a different secondary arc signature. Initial tests showed the algorithm was unsuccessful. The algorithm tended to indicate the arc had extinguished early, before the actual safe reclose time. It was assumed that the neural networks required retraining with waveforms generated from a 400kV line. In fact, this was not the case. Rather, the higher energies involved in the 400kV arc were saturating the hard limiter in the input scaling, falsely indicating an arc extinguish condition. When the secondary system input voltage was scaled to half that of the 132kV system, the algorithm was entirely successful in 100% cases: 110 transient fault cases and 40 permanent fault cases. (The distance resolution of the transient cases was 2 km spacing and PI section line models were used where necessary. The permanent cases were 10 km resolution with resistances of 0.00001, 2, 50 and 100 Ω). The results show that the scaling stage is just as important as the neural networks and may therefore be the best place to concentrate future development. It also suggests it may be possible to commercialise the algorithm for all transmission voltages, without extensive network training, although further testing is necessary to qualify this.

7-6 Conclusion

This chapter documents the hardware and software used in real time testing. An extensive results section shows the adaptive autoreclosing algorithm to be 100% successful when deployed on the system including the wind farm model. Since the wind farm model was not used in the training cases, it confirms that the algorithm on this system is not affected by wind generation, especially by poor power quality. It does not extensively show the performance of the algorithm on other systems however, and this is an important area for further investigation as it has bearing on the cost/benefit of commercialising such an algorithm. The test section also discusses the possibility of inter-circuit coupling and three phase adaptive autoreclosing, but the secondary arc is too short on this particular system to facilitate this.

C - Conclusion

C-1 Introduction

This closing chapter presents a summary of the work completed. Main conclusions are discussed, followed by future avenues of enquiry, concluding with some closing comments.

C-2 Thesis Summary

In chapter 1, the basic concepts of power system protection were covered. Beginning from a historical perspective, progress in secondary system hardware was discussed; from electromechanical relays, through to modern digital and numerical units. Overcurrent, differential and distance elements, together with symmetrical components were covered, but particular emphasis was made on the uniqueness of high frequency, transient based protection. This is pertinent to this thesis since adaptive autoreclosing schemes are usually a form of transient based protection. Fault types and causes of faults were discussed. The final section offered a detailed look at the practice of autoreclosing. This chapter does not present any novelty as it covers well known concepts, but is necessary to put the thesis in context.

Chapter 2 goes on to present a detailed literature review on adaptive autoreclosing. Principally this is diagnosing whether a fault is transient or permanent. This relies on detecting the arcing signature in the transient fault, and lack thereof in the permanent case. The adaptive autoreclosing concept was extended to provide optimal stability by closing at the prefault rotor angle, also in fast reclosing, the importance of torsional forces on local generators is was highlighted. A number of AI and signal processing techniques from the literature that achieve this were discussed. The most important work was that by Fitton et all, and is based on frequency transforms and a multi layer perceptron. The chapter concluded by identifying the literature gap regarding the increasing impact of wind farms and specifically, power electronics used in power conversion. This was identified as the central question to be addressed by this thesis.

Chapter 3 presented a self-contained study into wind farm transients, specifically autoreclosure. The study uses DIgSILENT's PowerFactory, to ascertain the differences in short circuit signatures on a 132kV system. An array of test cases were conducted on the parameters known to affect the short circuit responses, comparing a control case with a DFIG model and a fully rated converter machine model. Wavelet transforms were used to provide high frequency analysis. This study did not include a realistic arc model, the emphasis being on the generation technology. The main conclusion was that the high frequency short circuit response is more affected by other parameters than the generating technology involved. Significantly, parameters that are unknown before the fault have more bearing on the signature than the type of generation. This led to the conclusion that AI methods that successfully overcome these uncertainties, should certainly cope with variation in generation technology.

The remainder of the thesis attempted to prove this assertion. Too many variables make it impossible to meaningfully quantify the effect of power electronics. It is possible however, to demonstrate that their impact is not significant by deploying adaptive autoreclosing on a system and observing its performance. A real world system would be the ultimate test of such a scheme. Since this is not feasible with the projects resources, a real time digital model is the next best option.

Chapter 4 therefore detailed the construction and design of such a model, based on a portion of the 132kV transmission system in Scotland. The initial section summarised Dommel's digital simulation method for lumped and distributed parameters and discussed the theoretical concepts relevant to the RTDS model. This was followed by a brief overview of RTDS hardware. The 132kV system was then documented, including the main parameters, loads and sources, instrument transformers, line construction and configuration, the wind farm model and fault arc modelling. Importantly, limitations were discussed, which also highlighted further avenues for investigation, discussed in the forthcoming section C-4.

Fault signatures are determined by a complex interplay of parameters. The adaptive autoreclosing problem has thus often been approached with soft computing methods that attempt to reproduce the human reasoning process. It was therefore necessary to digress and cover the AI computing techniques most applicable to electric power

systems. Chapter 5 presented an overview of key areas of AI, with the cautious acumen required when evaluating these concepts from a protection and control perspective. Special emphasis was given to the most relevant techniques in adaptive autoreclosing. These are neural networks, fuzzy logic and genetic algorithms. The chapter concluded with a discussion of how these are best applied to autoreclosing.

The following chapter dealt with the development of the autoreclosing algorithm based on fault waveforms obtained from the system. The primary arcing period was observed to be too close to a permanent resistance to make a diagnosis pre circuit-breaker operation. Therefore, three phase breakers must rely on observing the secondary arc caused by inter-circuit coupling between the isolated and healthy circuit. Harmonic emissions based on ER G5/4 were examined over the secondary arcing period using 3D spectrographic plots. These were compared with the effect of the CVT. When the decibel conversion was omitted, both effects were shown to be negligible against the variation between transient and permanent faults. The plots also showed a sharp attenuation in frequencies above 300Hz, and so the algorithm does not make use of any frequency higher than this. The stability of the system was examined by plotting the rotor angle during a single phase to ground fault. Even whilst the fault persists, the system does not loose synchronism. So transient stability, whilst important for autoreclosing in some systems, is not relevant to this 132kV line.

The remainder of chapter 6 described, in detail, the algorithm itself. The algorithm is based heavily on the Fitton et al technique, since this has been proven on real world systems and documented in the literature. However, the important innovation is the use of two parallel ANNs for different purposes. Fault type detection is a separate problem space to arc extinction detection, and the algorithm exploits this to give greater robustness. The time domain response of both neural networks was shown, along with the logic in interpreting neural network output.

Finally, chapter 7 discusses the hardware and software used to demonstrate the algorithm in real time. The hardware includes the RTDS, a bespoke-built interfacing enclosure, a workstation, and the X3-SD, a dedicated A/D data capture unit. A software program supplied with the X3-SD is adapted to include the algorithm with C++ code, which is executed on the workstation's CPU. The algorithm is shown to be

feasible for real time deployment. Results are presented that show the algorithm to be 100% successful in fault diagnosis and intelligent reclosure. The secondary arc due to inter-circuit coupling is typically not sustained for long enough to make a robust diagnosis.

C-3 Discussion

The central question in this thesis maybe succinctly stated; How do wind farms affect neural network based adaptive autoreclosing algorithms? This question is addressed through two investigations involving different software and systems, including a real time simulation of a real world system. Within the scope of these models, the weight of evidence would suggest that wind farms do not significantly affect adaptive autoreclosing. When focussing on harmonic emissions, the CVT model is shown to have just as much influence in the frequencies used in diagnosing the secondary arc. This is shown in the spectrogram plots in chapter 6, figures 6.4 – 6.11. The difference between the permanent fault and the secondary arc is great enough to overcome any variation in the system configuration. The technique is given added robustness by using two neural networks, dedicated to specific tasks, each giving a crisp Boolean outcome.

The question is only conclusively proved in the case when the networks are specifically trained for the overhead line configuration on which they are to be deployed. Further work is therefore required before such techniques may be commercialised, and this is discussed in the next section.

A number of secondary conclusions emerged whilst pursuing the central question:

- On the 132kV system, inter-circuit coupling does not drive a secondary arc long enough to make three phase adaptive autoreclosing feasible. (Chapter 7).
- The ANN is still the most applicable AI technique to adaptive autoreclosing.
- In power system protection, neural networks are most robust when assigned to Boolean outcomes.

- The robustness of neural network algorithms is increased by assigning ANNs to specific, well defined tasks, and then parallelising their execution.
- The investigation gives useful experience to researchers in transient based protection, and protection in general. How might one investigate short circuit signatures on wind farms? To what extent are wind farms likely to affect signatures in the main grid?
- Ferro-resonance is important in single phase autoreclosing, in both power transformers (chapter 3) and in CVTs after reclosure (chapter 7).

In addition, some further work outcomes should be highlighted. The typical two-bus system model is no longer sufficient to serve innovation in power system protection. Smart grid issues such as power quality, distributed generation and adaptive protection demand a more complex real time model. The RTDS is therefore an excellent research tool for such purposes. The potential of such a tool should be maximised by those institutions fortunate enough to own one. The real 132kV system pushes the available RTDS system to its full limits. However, it may be later augmented when more hardware becomes available due to the RTDS's modular upgrading capability. Even if this system is not used in future, it is hoped this work serves as an example of what can be done.

The X3-SD and workstation combine to make an extremely versatile and powerful real-time pseudo-relay platform. When used in conjunction the RTDS, they may be used in the development and testing of any IEC 61850 compliant relay. RTDS technologies have developed a new GTNET card that broadcasts to Ethernet switch at IEC 61850 rates of 80 samples per system cycle, and the X3-SD is capable of receiving this digital input. The A/D feature of the X3/SD also supports applications for legacy substations that require analogue inputs. The secondary system computing power available is considerable: with the X3-SD's on board FPGA and a multi-core PC, demanding applications may be developed and tested in conjunction with the primary system in real time. Multithreaded language may be written to mimic new IEDs with more than one processor. The decoupling of the secondary and primary system frees up RTDS resources, and gives an extra level of real world credibility to research output.

Research output from academia often fails at the development stage because a number of real world issues have not been considered. This is usually because it is not feasible to build and test a prototype. This stage is crucial for problems to be highlighted and addressed. Matlab's real time workshop and emdedded real time encoder, used in this work, greatly speeds up the prototyping process. Very complex algorithms may be developed in simulink and then directly written in C++ code without the need for expert programming skills. Clear entry points make it relatively straight forward to attach inputs for real time execution. It is hoped that in future, a similar approach may be common practice in academia, such that research output is relevant and useful to manufacturers.

C-4 Further Work

Perhaps the most important area for further work is testing the developed algorithm on different system configurations. This gives a better impression on the scope for commercial development. It is suspected that manufacturers are unlikely to be interested in a device that only works for a single system or requires onerous calibration. However, as mentioned in chapter 7, initial tests on a 400kV line show the 132kV algorithm to be 100% successful, once input scaling is accounted for. In all likelihood, for maximum robustness the training data for the ANNs would need to be extended to encompass the range of variation in the systems on which they were to be deployed. This may be tackled from a commercial perspective by including 'pre-canned' networks that are trained to recognise secondary arc presence and extinction for different system configurations. In this case, installation simply involves setting the parameters of the overhead line on which they are to be deployed. Given the wide range of parameters, a software program would need to be authored to generate an exhaustive training set of waveforms for each network. If an RTDS system is not available, this can be achieved with the ATP-EMTP software, or similar.

This PhD project was funded by the EPSRC within the FlexNet workgroup of the Supergen research consortium. FlexNet is a multidisciplinary research team focussing on future electrical network technologies [89]. A key work stream in FlexNet is the showcase and validation of research outputs. In collaboration with the University of

Strathclyde, a showcase of the algorithm is planned at a later date. This will take place in conjunction with the Univeristy of Starthclyde's research outputs in Wide Area Protection Schemes (WAMS). At time of writing, this is likely to involve an RTDS model of a double circuit 400kV line. The autoreclosing relay will be deployed on circuit A and a quadrature booster will be on placed the adjoining circuit. It is anticipated that the adaptive autoreclosing algorithm will demonstrate, in real time, a tangible benefit to the system and in particular better preserve the function of the phaseshifting transformer. A paper reporting research outcomes is planned at a later date, co-authored with the University of Strathclyde.

In terms of wind farm simulation, it would be useful, as mentioned in chapter 4, to extend the model. The DFIG wind farms are represented by a single machine. Real time simulation of the true number of machines would require an extremely large RTDS, but is technically possible. It is unlikely however, that any single project would have access to such a resource. This is more realistically achievable through ATP simulations, and these may even serve to confirm the fidelity of a single machine representation for future real time studies. Particularly weak points in the current model are the lack of a cabled collector system and using just a single step up transformer, and these may be addressed with a relatively modest extra hardware resource.

It is unlikely, on economic grounds, that UK utilities will upgrade to single pole tripping facility just so that fast autoreclosing is possible. This is because the UK system is very strong and interconnected, and lines currently operate well below their transient stability limits. It therefore may be useful to identify critical double circuit lines that sustain an appreciable secondary arc following a three-phase trip. This is likely to occur on longer lines at higher voltage levels: both factors that increase mutual coupling. A particularly important area here would be to consider the forces imparted by fast reclosure. This also applies to wind farms: what are the mechanical stresses on the turbines imparted by autoreclosing? Presumably less - in wind installations they are more numerous than turbines in conventional plants - and they are somewhat decoupled by power electronics. Perhaps more significantly, future research should be directed towards the effect of fast autoreclosure on the converters themselves.

The real time development environment is possibly the most useful work outcome from this project. As discussed in the previous section, it provides a powerful and versatile environment for protection and control development and testing. Whilst large companies no doubt have more extensive hardware, this project shows how this may be achieved relatively inexpensively within an academic environment.

C-5 Epilogue

This concluding section is the author's own opinion and should therefore be taken with the proverbial *pinch of salt*. It is included in the main body of the thesis in the hope that key stakeholders will find the author's perspective useful.

21st century society is fast emigrating to the digital domain and this discipline is no exception. Due to its complexity, modern power system analysis is almost exclusively conducted on computers with specialist software. Some engineers are therefore becoming detached from the mathematics that underpin such programs. It is somewhat analogous to a mechanic opening the bonnet of a car and plugging it into a computer. The mechanic is increasingly detached from the underlying processes. There is no kinaesthetic interaction with the car's systems. A friendly and convenient user interface does make life easier, but can also lead to a "plug in the values and press a button" culture. It is important that engineers continue to be mindful of what is really going on under the bonnet and retain an understanding of the underlying processes. Results must be crosschecked on other software, allowing one to differentiate from user error, computational instability and actual power system phenomena so that output from computers is meaningful and accurate.

The other effect of this is a detachment from the physical scale of these systems. Electric power systems are the largest machines ever constructed by humankind. Just a few hundred mAs across the heart can be fatal. On power system scales, electricity is an awesome force that demands respect. If all researchers shared this safety critical ethos, a great deal more research output would be useful to the industry. All too often good ideas fail in an industrial context due to unnecessary complexity and lack of transparency. This is not an excuse for laziness either. An *overly* conservative attitude from the ageing work force may sometimes be attributed to a yearning for the quiet

life. Yet a sustainable generation mix demands a great deal from power systems and those responsible for them. The overwhelming opinion amongst climate scientists is that humanity will not have another chance at this. To paraphrase this point - the old adage: "Keep it simple, stupid" is often cited without much thought. The author much prefers Einstein's "Keep things as simple as possible… but no simpler". If nothing else, at least this makes power an interesting sector to be in again.

This PhD studentship involved a short industrial placement of around a month. The author found this extremely useful and believes this should be a compulsory element to any engineering PhD. Whilst on the placement; working at 'a very well known utility', the author was, however, rather shocked to discover a lot more accounting took place than engineering. Apparently, only 30% of the workforce were actually involved in engineering. The consequence seemed to be the economic pruning of a system that was nearing the end of its appreciable lifespan.

The fat of the system has been sufficient for some 60 years, but now, technical constraints are going to get extremely tough. These issues cannot be governed solely by short-term economic drivers, especially when these tend to be set by humanities and arts graduates in Whitehall, or ultimately financers in the city. The technical event horizon in this sector can be up to 50 years, and should be de-sensitised from stochastic economic cycles. If the privatised industry is left to its own devices, it will never deliver the smart grid infrastructure. If a watertight business case must be constructed for every innovation, progress will be too slow. The scale of the response should be proportional to the problem. Climate change therefore demands that society globally is placed on a war footing. Forty years ago, a decade long Apollo program culminated with people on the moon. Surely smart grids are achievable today; after all: it's not rocket science.

Politicians worry about the next election but Statesmen worry about the next generation. 'Statespeople' is a more appropriate term: addressing the gender imbalance in engineering would be a huge benefit to society. More Statespeople and less politicians are required in the energy sector. So where did the engineers go? Certainly not to the regulator it seems. There is a critical shortage of engineers, globally, and particularly in the UK. This is probably down to a lack of perception

about what engineers do and what their worth is to society. Engineers and scientists must stand up for themselves, and stand up for the truth that they strive to reveal. It should not be said that they work with *scientific evidence*, the word *evidence* is more effective and just as accurate. Public engagement is necessary to demystify this discipline and most importantly, instil a passion for science and engineering amongst the next generation. It is vital to the survival of humanity.

References

[1] ITER. *ITER Official Website*, 2007, [cited 2008], available at: www.iter.org

[2] BERR: *Department for Business Enterprise and Reform website*, 2008, [cited 2008], available at: www.berr.gov.uk/whatwedo/energy/sources/renewables/index.html

[3] BWEA, *British Wind Energy Association Official Website*, 2008, [cited 2008], available at: www.bwea.com

[4] Areva T&D (2008), *The Network Protection and Automation Guide*, available at www.alstom.com/grid/NPAG

[5] IEC 61850, IEC official website, accessed 2011, available at www.iec.ch/smartgrid/standards

[6] S. H. Horowitz and A. G. Phadke, *Power system relaying*, 3rd ed, 2008, Chichester, John Wiley, 331 p.

[7] Z. Q. Bo, *et al.*, *Transient based protection for power transmission systems*, 2000 Winter Meeting of the IEEE Power Engineering-Society, Singapore 2000, p. 1832-1837.

[8] H. W. Dommel, *Electromagnetic Transients Program Reference Manual (EMTP Theory Book)*. Portland, Bonneville Power Administration, 1986.

[9] C. L. Fortescue, *Polyphase Power Representation by Means of Symmetrical Coordinates*, Transactions of the American Institute of Electrical Engineers, 1918, vol. XXXIX, p 3.

[10] J. L. Blackburn, *Symmetrical components for power systems engineering*. 1993, New York, Marcel Dekker.

[11] Soares, A., M.A.O. Schroeder, and S. Visacro, *Transient voltages in transmission lines caused by direct lightning strikes*, IEEE Transactions on Power Delivery, 2005. **20**(2): p. 1447-1452.

[12] IEEE Power Engineering Society, *C37.104-2002: IEEE guide for automatic reclosing of line circuit breakers for AC distribution and transmission lines*, New York, 2003.

[13] Gillies, D. A., *Operating experience with 230-kV automatic reclosing on the Bonneville power administration system*, AIEE Transactions, Part III: P-A-S, 1954, vol. 75, p.1692–1694

[14] G. D. Rockefeller, et al., *Adaptive transmission relaying concepts for improved performance,* IEEE Transactions on Power Delivery, 1988, vol. 3, p.1446-1458.

[15] A. G. Phadke, et al., *Adaptive Automatic Reclosing*, presented at the CIGRE, Report No. 34-204, Paris, 1990.

[16] D. S. Fitton, et al., *Design and implementation of an adaptive single pole autoreclosure technique for transmission lines using artificial neural networks,* IEEE Transactions on Power Delivery, 1996, vol. 11, p. 748-756,

[17] R. K. Aggarwal, et al., *Neural-Network based adaptive single-pole autoreclosure techniques for EHV transmission systems*, IEE Proceedings-Generation Transmission and Distribution, vol. 141, p. 155-160, 1994.

[18] I. K. Yu and Y. H. Song, *Development of novel adaptive single-pole autoreclosure schemes for extra high voltage transmission systems using wavelet transform analysis*, Electric Power Systems Research, vol. 47, p. 11-19, 1998.

[19] M. A. El-Hadidy, et al., *Using neuro-wavelet technique for adaptive single phase autoreclosure of transmission lines*, in 39th International Universities Power Engineering Conference (UPEC 2004), Bristol, UK, 2004, p. 684-688.

[20] H. Khorashadi-Zadeh, et al., *Transmission Line Single Phase Auto Reclosing Scheme Based on Wavelet Transform and Adaptive Fuzzy Neuro Inference System,* in 39th North American Power Symposium, Las Cruces, NM, 2007, p. 43-48.

[21] Frimpong, E.A., *Intelligent circuit breaker control for transmission systems employing single pole/three phase autoreclosure*, in *Electronic and Electrical Engineering*. 2006, *[MSc Dissertation]*, University of Bath, Bath.

[22] V. Terzija, *Efficient distance protection and adaptive autoreclosure numerical algorithm*, 2001 IEEE/PES Transmission and Distribution Conference and Exposition, 2001, vols. 1 and 2: *Developing New Perspectives*, p. 669-674.

[23] V. V. Terzija and Z. M. Radojevic, *Numerical Algorithm for Adaptive Autoreclosure and Protection of Medium-Voltage Overhead Lines*, IEEE Transactions on Power Delivery, 2004, vol. 19, p. 5,

[24] Z. M. Radojevic and J. R. Shin, *New digital algorithm for adaptive reclosing based on the calculation of the faulted phase voltage total harmonic distortion factor,* IEEE Transactions on Power Delivery, 2007, vol. 22, p. 37-41.

[25] I. P. Gardiner and J. Ramsden, *On Site Experience of an Adaptive Autoreclose Relay for ovehead lines,* IEE conference publication on developments in power system protection, Nottingham,1997, p. 377-380.

[26] R. K. Aggarwal, et al., *Adaptive three-phase autoreclosure for double-circuit transmission-systems using neural networks*, 2nd International Conference on Advances in Power System Control, Operation & Management, 1993, vols. 1 and 2, p. 389-392.

[27] S. P. Websper, et al., *An investigation into breaker reclosure strategy for adaptive single pole autoreclosing*, IEE Proceedings-Generation Transmission and Distribution, 1995, vol. 142, p. 601-607.

[28] B. H. Zhang, et al., *The design and application of an optimal reclosure technique for transmission lines*, North American Power Symposium, Montana State University, Bozeman, 1995.

[29] Z. Q. Bo, et al., *New concept in transmission line reclosure using high-frequency fault transients*, IEE Proceedings-Generation Transmission and Distribution, vol. 144, p. 351-356, 1997.

[30] A. T. Johns, et al., *Study of turbine-generator torsional oscillations with particular reference to adaptive autoreclosure*, Tencon '93: 1993 IEEE Region 10 Conference on Computer, Communication, Control and Power Engineering, vol. 5, p. 133-136, 1993.

[31] Y.-J. Lee, et al., *Development of Auto-Reclosing Algorithm Using Multi Agent System*, Developments in Power System Protection, 2008, Glasgow, 2008.

[32] Song, Y.H., R.K. Aggarwal, and A.T. Johns, *Digital simulation of fault arcs on long-distance compensated transmission-systems with particular reference to adaptive autoreclosure*, European Transactions on Electrical Power Engineering, 1995, vol. 5, p. 315-324, Wiley

[33] T. Ackermann, *Wind power in power systems*, 2005, Wiley Chichester, West Sussex, UK

[34] P. Ledesma, J. Usaola, and J. L. Rodriguez, *Transient stability of a fixed speed wind farm*, Renewable Energy, 2003, vol. 28, p. 1341-1355.

[35] F. Mei and B. Pal, *Modal analysis of grid-connected doubly fed induction generators*, IEEE Transactions on Energy Conversion, 2007, vol. 22, p. 728-736.

[36] A. S. Neto, et al. *Dynamic analysis of grid connected wind farms using ATP*, IEEE 36th Power Electronic Specialists Conference (Pesc), 2005, Vols 1-3, p. 198-203.

[37] G. Saccomando, J. Svensson, and A. Sannino, *Improving voltage disturbance rejection for variable-speed wind turbines*, IEEE Power Engineering Society Summer Meeting, 2002, Vols 1-3, Conference Proceedings, p. 502-502.

[38] Hansen, L.H., et al., *Conceptual survey of Generators and Power Electronics for Wind Turbines*, 2001, Risø National Laboratory, Roskilde, Sweden

[39] Carlsson, A, *The Back-to-back Converter - Design and Control, [PhD thesis]*, Lund University, Sweden, 1998

[40] Kim, C.H. and R. Aggarwal, *Wavelet transforms in power systems - Part 1 - General introduction to the wavelet transforms* Power Engineering Journal, 2000, 14(2): p. 81-87.

[41] Kim, C.H. and R. Aggarwal, *Wavelet transforms in power systems Part 2 Examples of application to actual power system transients.* Power Engineering Journal, 2001, 15(4): p. 193-202.

[42] Daubechies, I., *Ten lectures on wavelets*. CBMS-NSF regional conference series in applied mathematics,1992, Philadelphia.

[43] BERR SEDGE (centre for Sustainable Electricity and Distributed Generation), *UK Generic distribution network*, 2004; [cited 2008], available at: www.sedg.ac.uk/ukgds.htm.

[44] Johns, A.T. and Aggarwal, R.K., *Digital simulation of faulted EHV transmission lines with particular reference to very high speed protection*, Proceedings of the IEE London, 1976 **123**(4): p. 353-359.

[45] Ramtharan, G., Jenkins, N., and Anaya-Lara, O., *Modelling and Control of Synchronous Generators for Wide-range Variable-speed Wind Turbines, Wind Energy*, 2007, vol 10, No. 3. p. 231-246

[46] Johns, A.T., et al, *Spectrum analysis of fault-induced transients for the development of protection equipment*, 2nd International Conference on Advances in Power System Control, Operation and Management 1993, Hong Kong.

[47] Morren, J. and S.W.H. de Haan, *Ridethrough of wind turbines with doubly-fed induction generator during a voltage dip*, IEEE Transactions on Energy Conversion, 2005. **20**(2): p. 435-441.

[48] Pena, R., J.C. Clare, and G.M. Asher, *Doubly fed induction generator using back-to-back PWM converters and its application to variable-speed wind-energy generation*, IEE Proceedings-Electric Power Applications, 1996. **143**(3): p. 231-241.

[49] Dommel, H. W., *Digital Computer Solution of Electromagnetic Transients in Single-and Multiphase Networks*, IEEE Transactions on Power Apparatus and Systems, 1969, vol.PA88 (4), p. 388-399.

[50] F. H. Branin, *Computer methods of network analysis*, Proc. of the IEEE, 1967, vol. 55, p. 1787-&, 1967.

[51] Semlyen, A. and Dabuleanu, A., *Fast and accurate switching transient calculations on transmission lines with ground return using recursive convolutions*, IEEE Transactions on Power Apparatus and Systems, 1975. **94**(2), p. 561-571.

[52] Marti, J., *Accuarte Modelling of Frequency-Dependent Transmission Lines in Electromagnetic Transient Simulations.* IEEE Transactions on Power Apparatus and Systems, 1982. **PAS-101**(1): p. 147.

[53] RTDS Technologies, *Real Time Digital Simulator Power System Users Manual*, Winnipeg, Canada, November 2006, www.rtds.com

[54] National Grid, *National Grid GB Seven Year Statement*, 2008, [cited 2009], available at www.nationalgrid.com/uk/sys_08

[55] Weedy B. M., and Cory B. J., *Electric power systems*, 4th ed, Chichester, Wiley, 1998.

[56] Maguire, T. and Giesbrecht, J., *Small Time-step (<2 microseconds) VSC Model for the Real Time Digital Simulator*, International Conference on Power Systems Transients, Montreal, Canada, 2005.

[57] Forsyth, P.A., Shearer, D. and Rydmell, D., *Testing Firing Pulse Controls for a VSC-based HVDC Scheme with a Real Time Timestep < 3 μs*, International Conference on Power Systems Transients, Japan, 2009.

[58] Fitton, D., *A Neural Network Based Adaptive Single Pole Autoreclose Technique, [PhD thesis]*, 1995, Department of Electronic and Electrical Engineering, University of Bath, Bath.

[59] Kojovic, L, Kezunovic, M., Skendzic, V., Fromen, C.W., and Sevcik, D.R., *A New Method for the CCVT Performance Analysis Using Field Measurements, Signal Processing and EMTP Modeling*, IEEE Transactions on Power Delivery, 1994, vol. 9, no. 4, p. 1907-1915.

[60] Johns, A.T., R.K. Aggarwal, and Y.H. Song, *Improved techniques for modelling fault arcs on faulted EHV transmission systems.* Generation, Transmission and Distribution, IEE Proceedings, 1994, 141(2): p. 6.

[61] Strom, A.P., *Long 60-cycle arcs in air*, Transactions of American Institute of Electrical Engineers, 1946, 65, p. 113-117.

[62] Giesbrecht, J., Ouellette, D.S. and Henville, C. F., *Secondary Arc Extinction and Detection - real and simulated*, DPSP, Glasgow 2008.

[63] Warwick, K., *et al.*, *Artificial intelligence techniques in power systems*, 1st ed, 1997, London IEE.

[64] Mitchell M., An Introduction to Genetic Algorithms, 1st ed, Cambridge, Mass.London: MIT press, 1996.

[65] Bahbah, A. G. and Girgis, A. A., *An Investigation On The Effect Of Line Reclosing On Transient Stability Assessment For Multi-Machine Power Systems*, IEEE PES Summer Meeting,1999.

[66] Wooldridge, M. J., *An introduction to multiagent systems*, Chichester: J. Wiley, 2002.

[67] Apostolov, A., *Multi-agent systems and IEC 61850*, Power Engineering Society General Meeting, 2006,Vols 1-9, pp. 2202-2207,

[68] Mathworks, *MATLAB Fuzzy Logic Toolbox Tutorial*, 2004, cited 2010 available at www.mathworks.com/access/helpdesk/help/toolbox/fuzzy/index.html

[69] Bansal, R. C, *Literature survey on expert systems applications to power systems (1990-2001)*, Engineering Intelligent Systems for Electrical Engineering and Communications, 2003, vol. 11, p. 103-112.

[70] Chang, C. S., et al., *Fuzzy expert system for fault diagnosis in power systems*, Engineering Intelligent Systems for Electrical Engineering and Communications, 1997, vol. 5, p. 75-81.

[71] Yeo, S. M., et al., *A novel algorithm for fault classification in transmission lines using a combined adaptive network and fuzzy inference system*, International Journal of Electrical Power & Energy Systems, 2003, vol. 25, p. 747-758.

[72] Wang H. and Keerthipala W. W. L., *Fuzzy-neuro approach to fault classification for transmission line protection*, IEEE Transactions on Power Delivery, 1998, vol. 13, p. 11.

[73] Jarventausta P., et al., *Using fuzzy-sets to model the uncertainty in the fault location process of distribution networks*, IEEE Transactions on Power Delivery, 1994, vol. 9, p. 954-960.

[74] Xu, W., et al., *Fault diagnosis of power transformers: Application of fuzzy set theory, expert systems and artificial neural networks*, IEE Proceedings-Science Measurement and Technology, 1997, vol. 144, p. 39-44.

[75] EL-Shimy, M., Badr, M.A.L, and Rassem, O.M., *Impact of large scale wind power on power system stability*, 12th International Middle-East Power System Conference, New York, 2008.

[76] Aggarwal, R., and Song, Y. H., Artificial neural networks in power systems part 1. General introduction to neural computing, Power Engineering Journal, 1997, vol. 11, p.129-134.

[77] Aggarwal, R., and Song, Y. H., *Artificial neural networks in power systems Part 2: Types of artificial neural networks*, Power Engineering Journal, 1998, vol. 12, p 41-47.

[78] Hopfield, J. J., *Neural networks and physical systems with emergent collective computational abilities*, Proceedings of the National Academy of Sciences of the USA, 1982, vol. 79, no. 8, p. 2554-2558.

[79] Kohonen, T., *Self-organized formation of topologically correct feature maps*, Biological Cybernetics, 1982, vol.43, p 59-69.

[80] Bouthiba, T., *Fault location in EHV transmission lines using artificial neural networks*, International Journal of Applied Math. and Computer Science, 2004, vol. 14, p. 69-78.

[81] Electricity Association, *Planning levels for harmonic voltage distortion and the connection of non-linear equipment to transmission systems and distribution networks in the United Kingdom*, 2001. available at www.harmonicsolutions.co.uk, cited 2011.

[82] Hensman G., *Connecting nonlinear loads to public electricity systems - a guide to Engineering Recommendation G5/4*, Power Engineering Journal, vol. 16, pp. 77-87, 2002.

[83] Mathworks, *Discrete Fourier Transform*, 2004, cited 2011, available at www.mathworks.com/help/techdoc/ref/fft.html

[84] RTDS technologies, *RTDS hardware manual*, unpublished, contact RTDS technologies, www.rtds.com

[85] Innovative Integration, *X3-SD manual*, available at www.innovative-dsp.com/cgi-bin/dlDocs.cgi?product=X3-SD, [cited 2011]

[86] Innovative Integration *Malibu library manual*, available at http://www.innovative-dsp.com/cgi-bin/dlDocs.cgi?product=X3-SD, [cited 2011]

[87] SisSoftware, *SisSoftware official website*, 2010, available at www.sisoftware.co.uk, [cited 2011]

[88] RTDS technologies, *RTDS tutorial manual*, unpublished, contact RTDS technologies, www.rtds.com

[89] FlexNet, *FlexNet official site*, 2009, available at www.supergen-networks.org.uk, [cited 2011]

Appendix 1

A1 – a) System model

The external grid

The following page shows a high-resolution print out of the system model in RSCAD draft. This shows the external grid. Fig 1.1 is an annotated low resolution key to this diagram. The Farr and Paul's wind farm busses must exist in a second subsystem so RTDS resources may be distributed appropriately. This subsystem is executed on another RTDS rack and connected by a travelling wave transmission line model.

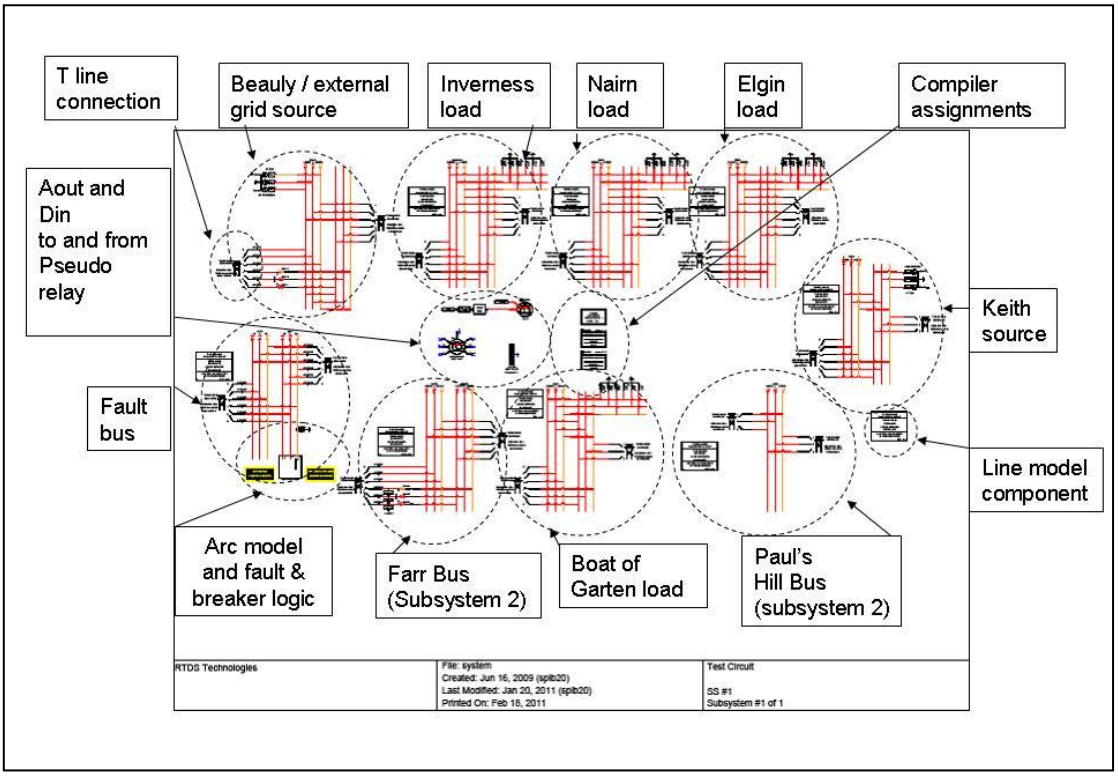
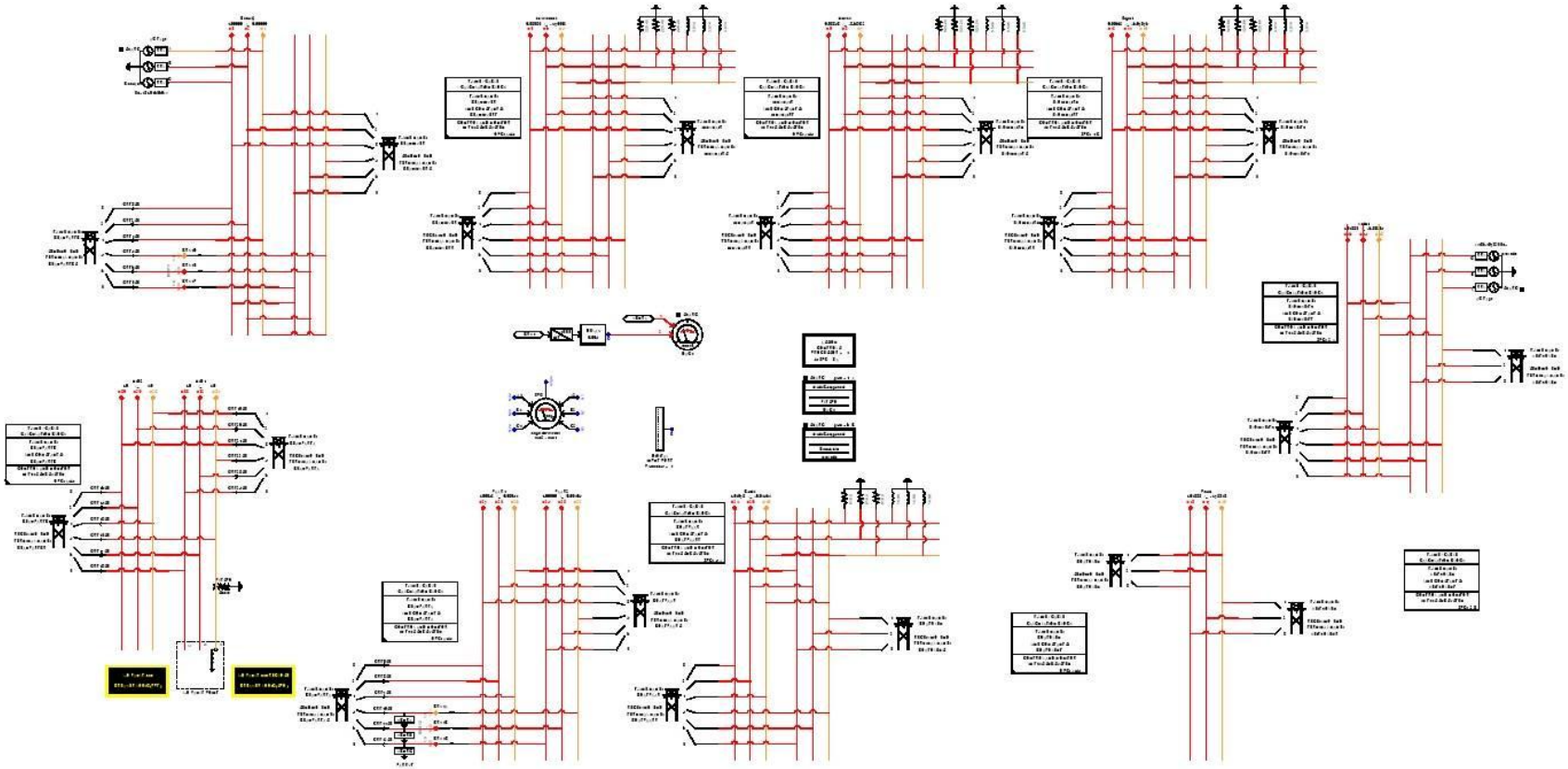


Figure A1.1: The system model in RSCAD draft



RTDS Technologies	File: system Created: Jun 16, 2009 (splb20) Last Modified: Jan 20, 2011 (splb20) Printed On: Feb 18, 2011	Test Circuit SS #1 Subsystem #1 of 1

Wind Farms

The two wind farms are executed in a separate subsystem in draft, meaning they are physically computed on a different RTDS rack. A single wind farm model is shown for illustrative purposes.

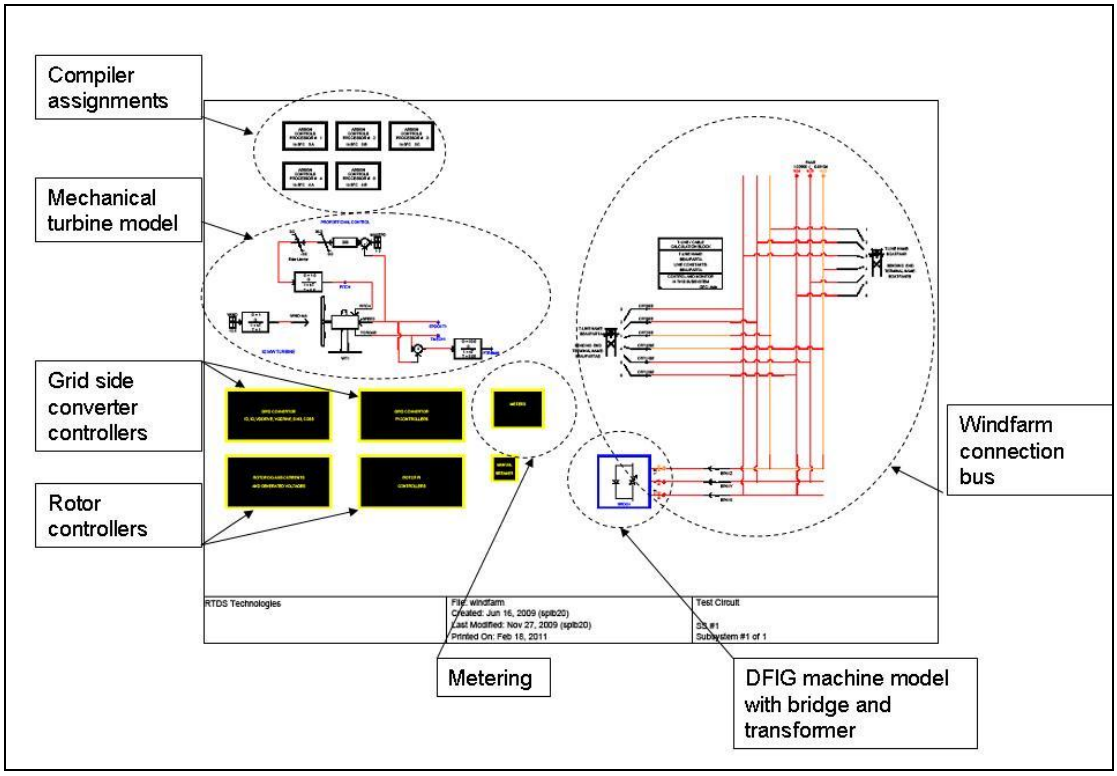
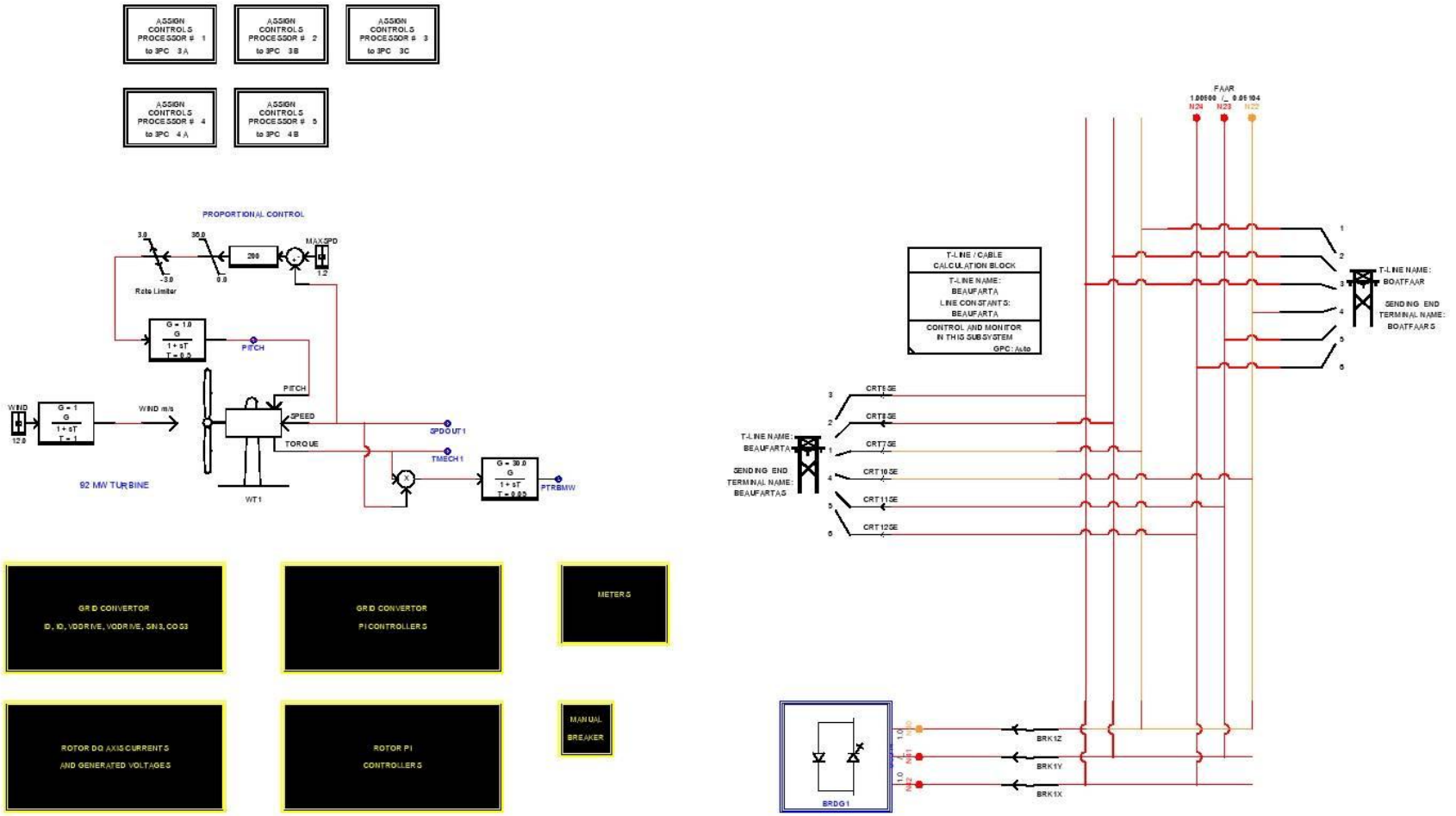


Figure A1.2: The windfarm model in RSCAD draft

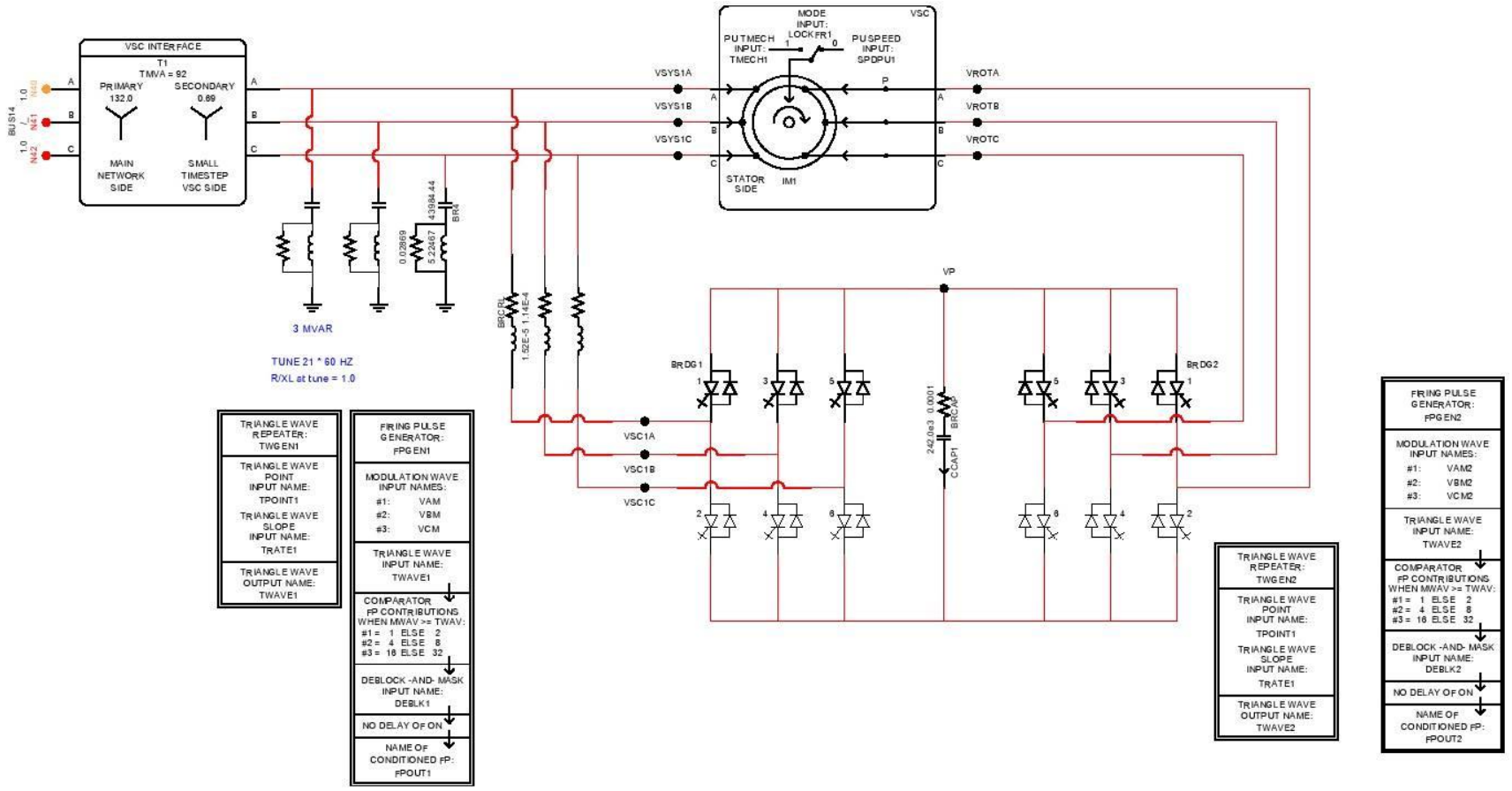


RTDS Technologies	File: windfarm Created: Jun 16, 2009 (splb20) Last Modified: Nov 27, 2009 (splb20) Printed On: Feb 18, 2011	Test Circuit SS #1 Subsystem #1 of 1

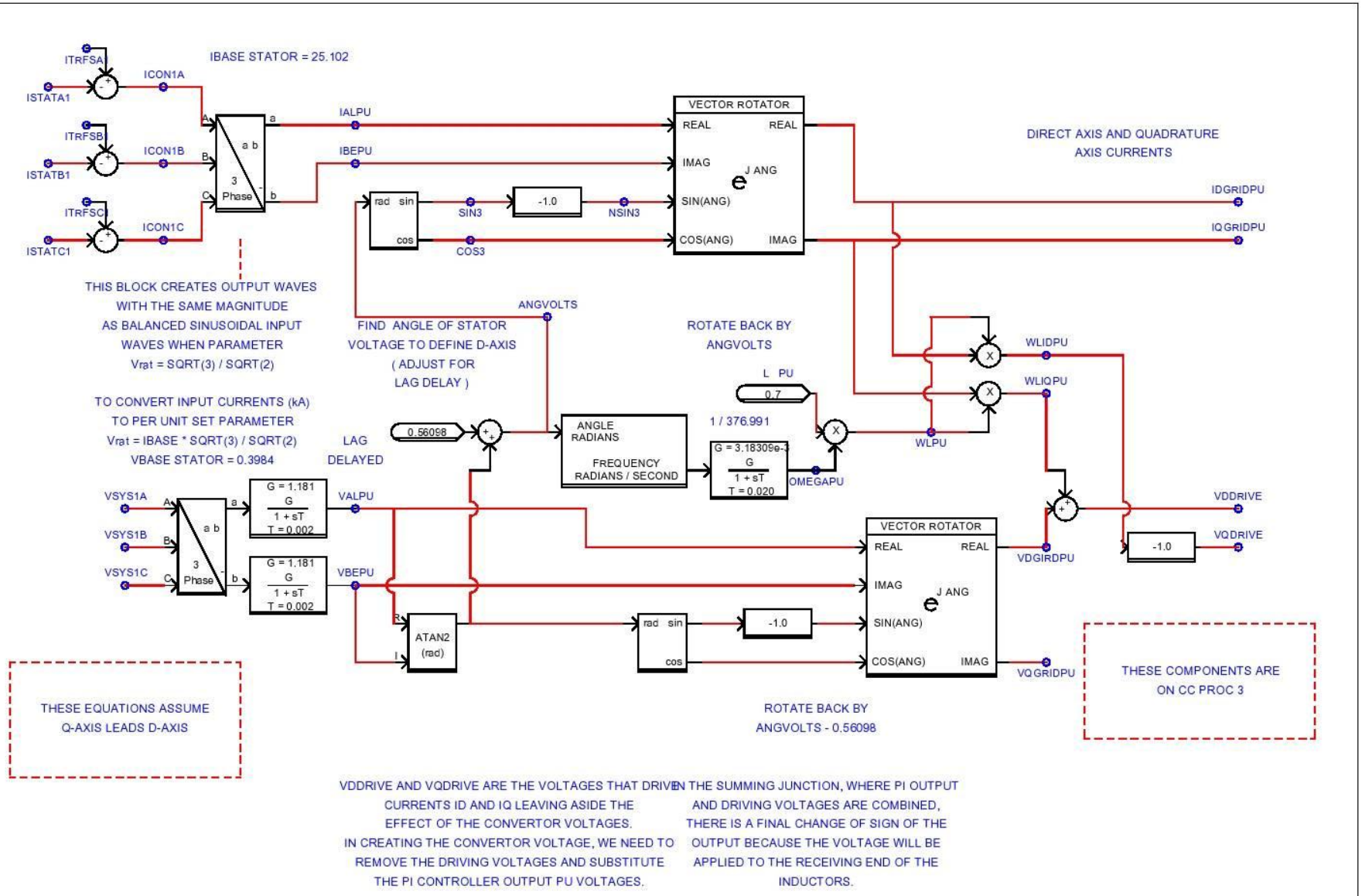
Wind farm control

It should be stressed that wind farm control is a complex sub-discipline of control engineering, the details of which are outside the focus of this project. The controls in the RTDS technologies DFIG model are designed to be manufacturer agnostic. The extent to which they resemble separate commercial examples may vary considerably. The important commonality with all DFIG machines, however, and pertinent to short circuit grid signatures, is the closely coupled control of the rotor and stator assembly together with the back-to-back converter. This is described in chapter 3. With advice from RTDS technologies, their turnkey model was augmented to fit the global system parameters, such as the frequency, filters, and the model power and apparent power. Screenshots are included for illustration purposes.

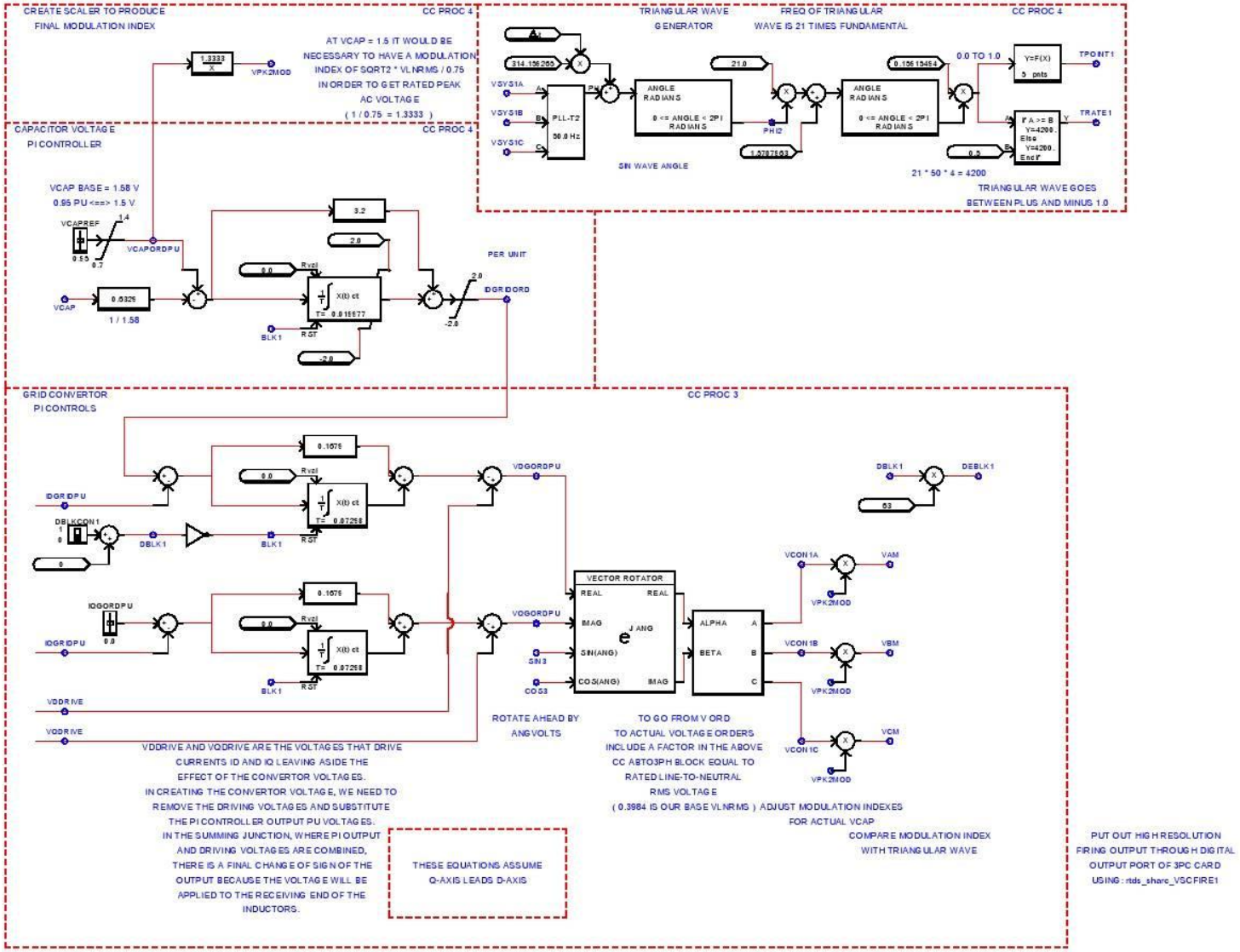
the bridge and the machine……..*199*
grid/stator side control……..*200*
grid/stator side PI-controllers……..*201*
rotor side control……..*202*
rotor PI- controllers……..*203*



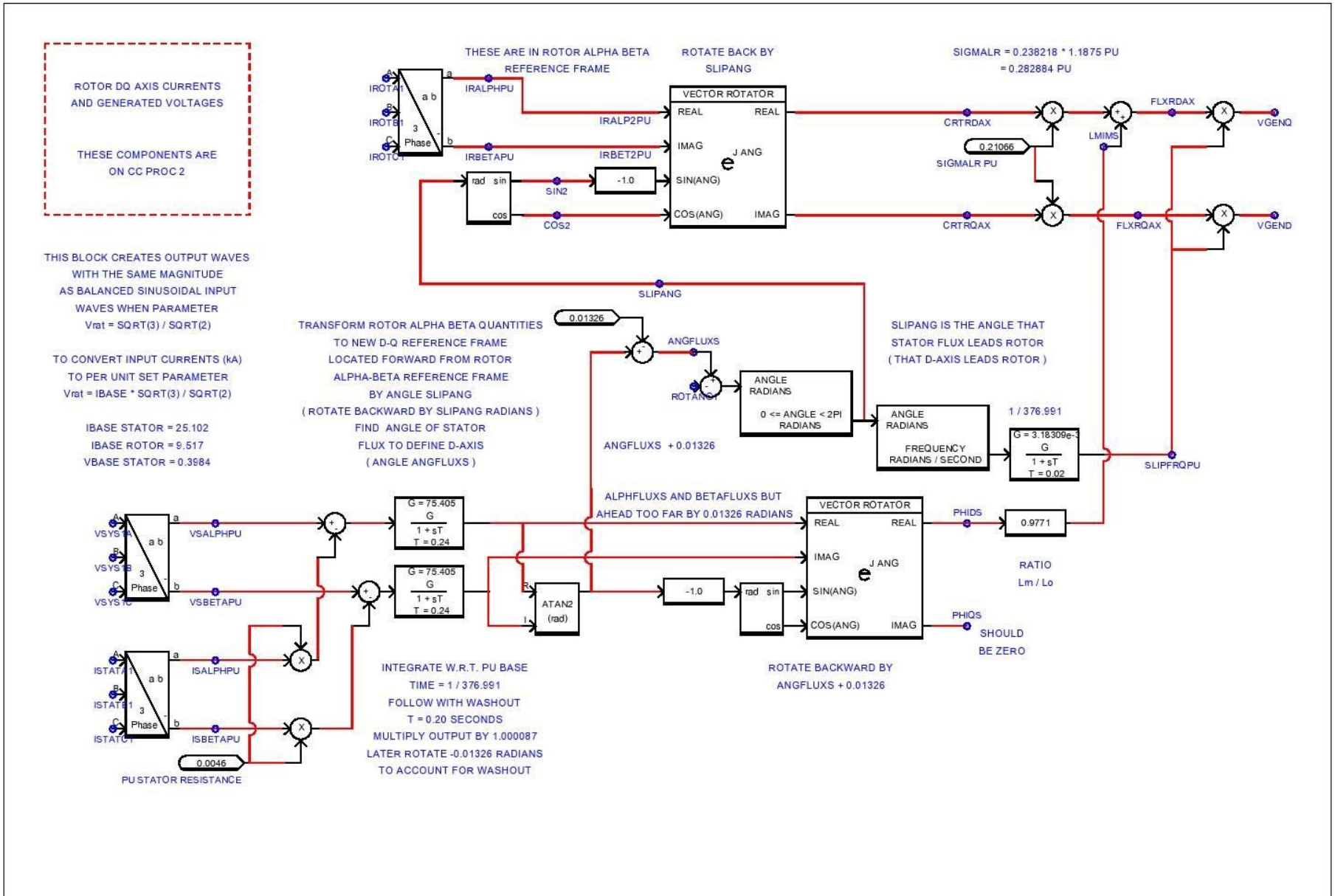
RTDS Technologies	File: windfarm Created: Jun 16, 2009 (splb20) Last Modified: Nov 27, 2009 (splb20) Printed On: Feb 18, 2011	Test Circuit TRANSFORMER AND LOAD SS #1 Subsystem #1 of 1



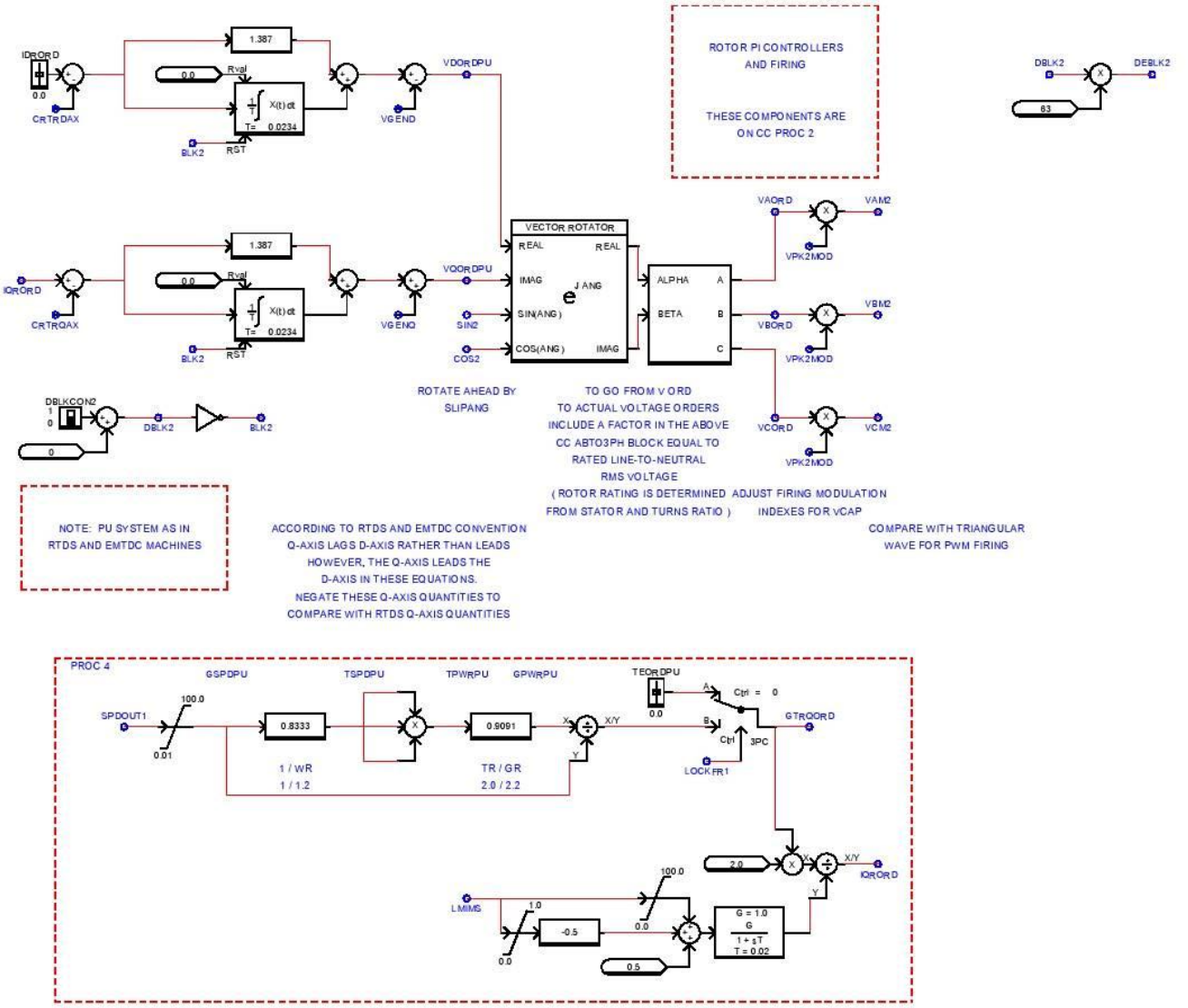
RTDS Technologies	File: windfarm Created: Jun 16, 2009 (splb20) Last Modified: Nov 27, 2009 (splb20) Printed On: Feb 18, 2011	Test Circuit GRID CONVERTOR ID, IQ, VDDRIVE, VQDRIVE, SIN3, COS3 SS #1 Subsystem #1 of 1



RTDS Technologies	File: windfarm Created: Jun 16, 2009 (splb20) Last Modified: Nov 27, 2009 (splb20) Printed On: Feb 18, 2011	Test Circuit GRID CONVERTOR PI CONTROLLERS SS #1 Subsystem #1 of 1



RTDS Technologies	File: windfarm Created: Jun 16, 2009 (splb20) Last Modified: Nov 27, 2009 (splb20) Printed On: Feb 18, 2011	Test Circuit ROTOR DQ AXIS CURRENTS AND GENERATED VOLTAGES SS #1 Subsystem #1 of 1



RTDS Technologies	File: windfarm Created: Jun 16, 2009 (splb20) Last Modified: Nov 27, 2009 (splb20) Printed On: Feb 18, 2011	Test Circuit ROTOR PI CONTROLLERS SS #1 Subsystem #1 of 1

A1 – b) RSCAD Runtime

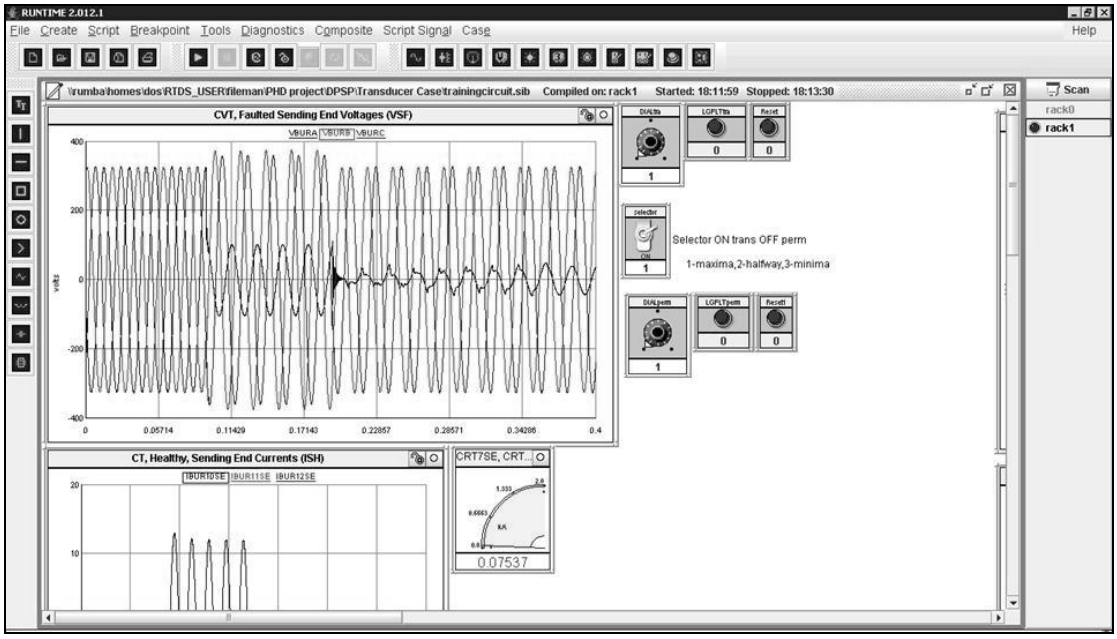


Figure A1.3: An example of RSCAD runtime canvas

Figure A1.3, an example canvas of the Runtime component of RSCAD. The draft case is compiled and executed in Runtime. Here, the user is able to make real time changes to the system and observe the response.

Appendix 2 – Real time C++ code

The SNAP example software supplied with the Innovative Integration X3-SD is a large and complex program. It is also proprietary to Innovative Integration so cannot be reproduced in its entirety for copyright reasons. The source and header files are included along with 'MALIBU' an extensive DSP and I/O C++ library. The user may extend SNAP software with custom applications in C++.

As far as the real time execution of the algorithm is concerned, there are three important sections in the source code program that have been augmented for the purposes of this project. Since the header files are not included, the code in this appendix is for illustration purposes only.

1) call to algorithm when packet stream received ……...205
2) entry points to algorithm……..206
3) algorithm main source file…….. 207
4) algorithm data and parameters…….236

1) The SNAP function `ApplicationIo::HandleDataAvailable`, declared in the header file ApplicationIO.hpp and defined in ApplicationIO.hpp, sends data to the algorithm when a packet arrives from the board.

```
void  ApplicationIo::HandleDataAvailable(PacketStreamDataEvent &
Event)
{
      // [proprietry code…]


       // Code for Bath
    int* pData = Packet.Data();
    int  Count = Packet.SizeInInts();

    // pass new samples to BathsAlg 1 sample at a time. The int is
//raw Adc samples,
    // so between -8,388,608 and +8,388,607.



      for( int n=0 ; n < Count ;)
      {
            inputtomodel = ( pData[n++] ); // the first sample is the
                                           // data
```

```
            breakerin = ( pData[n++] );     // the second sample is
                                            // the breaker status

            if (breakerin < 0 && firsttime == true)
                              // the breaker status and first zero
                              // crossing

            {breakerstatus = 1;
            firsttime = false;}

            if (breakerstatus == 1)
            {cycle++;};//} 1 at 20kHz power cycle for CB to attenuate
                        //20/50

            if (cycle > 400){ // 1 cycle at 20kHz (power cycle for CB
                              to attenuate) = 400 samples

            bool safetoreclose = false;

            int outputfromANN =
ApplicationIo::senddatatomodel(inputtomodel, breakerstatus); // send
// data to model


            if(outputfromANN == 1)
                  safetoreclose = true; // flag breaker if alg
                                              inficates it


            if(safetoreclose)
            Module.Dio().DioPortData().Value(14);} // set D/O to
                                              //    logical 1
                              //    (masked 16 bit signal, inverted
                              //     so 15 = 0, 14 = 1 etc

            };

      // End code for Bath

      }
```

The variables within this function are declared as class members by the header file that declares the ApplicationIO class. They are initialised when the stream button is pressed in the snap application.

2) The entry point to the functions are within the ertmain.cpp. The interaction of the time steps are controlled when this function is called although many of these variables must be declared in their header files and initialised outside the function. The rt _onestep function returns a default 100 for every 7 of 8 samples it receives, but at the

lower sample rate, every 1 of 8, it executes the algorithm and either returns a 1 or a 0 to indicate the desired reclose circuit breaker status.

```
double rt_OneStep(double inputtomodel, double breakerstatus)
{
  /* Disable interrupts here */

 double breakerfloat;

      if (breakerstatus > 0)
      breakerfloat = 1.0
      else breakerfloat = 0.0;

      double modelout=100;
inputtomodel = inputtomodel * 1.9563e-004;//scale the value to give
the magnitude of the CVT waveform

schemefinal_step0(inputtomodel, breakerfloat);

taskCounter[1]++; //this cycles round between 1 and 8
                  //so that the alg may share the two sample rates

      if (taskCounter[1]== 1) {
      OverrunFlags[1]++;

      modelout = schemefinal_step1(breakerfloat); //scale the value

      return modelout;
      }

 if (taskCounter[1] == 8); reset counter
      taskCounter[1] = 0;
}
```

3) The bulk of the algorithm resides in the main source file associated with the project. This includes the STFFT, the neural networks and the associated logic and the passing of signals between these elements.

```
/*
 * File: schemefinal.cpp
 *
 * Real-Time Workshop code generated for Simulink model schemefinal.
 *
 * Model version                        : 1.79
 * Real-Time Workshop file version      : 7.2  (R2008b)  04-Aug-2008
 * Real-Time Workshop file generated on : Thu Jan 13 15:14:25 2011
 * TLC version                          : 7.2 (Aug  5 2008)
 * C/C++ source code generated on       : Thu Jan 13 15:14:27 2011
 */

#include "schemefinal_capi.h"
#include "schemefinal.h"
#include "schemefinal_private.h"
```

```
#include <iostream>
#include <fstream>

using namespace std;


// added scalar function (required for header file compatability
//issues)


MWDSP_IDECL void MWDSP_CopyScalarICs( byte_T       *dstBuff,
                          const byte_T *ICBuff,
                          int_T         numElems,
                          const int_T   bytesPerElem )
{
    while (numElems-- > 0) {
        memcpy( dstBuff, ICBuff, bytesPerElem );
        dstBuff += bytesPerElem;
    }
};

/* Block signals (auto storage) */
BlockIO_schemefinal schemefinal_B;

/* Block states (auto storage) */
D_Work_schemefinal schemefinal_DWork;

/* Real-time model */
RT_MODEL_schemefinal schemefinal_M_;
RT_MODEL_schemefinal *schemefinal_M = &schemefinal_M_;
void MWDSPCG_FFT_Interleave_R2BR_D(const real_T *x, creal_T *y, const
int32_T
  nChans, const int32_T nRows)
{
  int32_T br_j;
  int32_T yidx;
  int32_T uIdx;
  int32_T j;
  int32_T nChansBy2;
  int32_T bit_fftLen;

  /* Bit-reverses the input data simultaneously with the interleaving
operation,
     obviating the need for explicit data reordering later.  This
requires an
     FFT with bit-rev inputs.
   */
  br_j = 0;
  yidx = 0;
  uIdx = 0;
  nChansBy2 = nChans >> 1;
  while (nChansBy2) {
    nChansBy2 = nChansBy2 - 1;
    for (j = nRows - 1; j > 0; j = j - 1) {
      y[yidx + br_j].re = x[uIdx];
      y[yidx + br_j].im = x[uIdx + nRows];
      uIdx = uIdx + 1;

      /* Compute next bit-reversed destination index */
      bit_fftLen = nRows >> 1;
```

```
      for (br_j = br_j ^ bit_fftLen; !(br_j & bit_fftLen); br_j =
br_j ^
           bit_fftLen) {
        bit_fftLen = bit_fftLen >> 1;
      }
    }

    y[yidx + br_j].re = x[uIdx];
    y[yidx + br_j].im = x[uIdx + nRows];
    uIdx = (nRows + 1) + uIdx;
    yidx = (nRows << 1U) + yidx;
    br_j = 0;
  }

  /* For an odd number of channels, prepare the last channel
     for a double-length real signal algorithm.  No actual
     interleaving is required, just a copy of the last column
     of real data, but now placed in bit-rev order.
     We need to cast the real u pointer to a cDType_T pointer,
     in order to fake the interleaving, and cut the number
     of elements in half (half as many complex interleaved
     elements as compared to real non-interleaved elements).
   */
  if (nChans & 1) {
    for (j = (nRows >> 1) - 1; j > 0; j = j - 1) {
      y[yidx + br_j].re = x[uIdx];
      y[yidx + br_j].im = x[uIdx + 1];
      uIdx = uIdx + 2;

      /* Compute next bit-reversed destination index */
      nChansBy2 = (nRows >> 1) >> 1;
      for (br_j = br_j ^ nChansBy2; !(br_j & nChansBy2); br_j = br_j
^ nChansBy2)
      {
        nChansBy2 = nChansBy2 >> 1;
      }
    }

    y[yidx + br_j].re = x[uIdx];
    y[yidx + br_j].im = x[uIdx + 1];
  }
}

void MWDSPCG_R2DIT_TBLS_Z(creal_T *y, const int32_T nChans, const
int32_T nRows,
  const int32_T fftLen, const int32_T offset, const real_T *tablePtr,
const
  int32_T twiddleStep, const boolean_T isInverse)
{
  creal_T *yCplx;
  int32_T nHalf;
  real_T twidRe;
  real_T twidIm;
  int32_T nQtr;
  real_T fwdInvFactor;
  int32_T iCh;
  int32_T idelta;
  int32_T ix;
  int32_T i2;
  real_T temp[2];
  int32_T k;
```

```
  int32_T kratio;
  int32_T istart;
  int32_T i1;
  int32_T j;
  real_T temp_0[2];
  real_T temp_1[2];
  real_T tmp;
  real_T tmp_0;
  yCplx = y;

  /* Signal Processing Blockset Decimation in Time FFT  */
  /* Computation performed using table lookup optimized for speed  */
  /* Output type: complex real_T */
  yCplx = &yCplx[offset];
  nHalf = (fftLen >> 1) * twiddleStep;
  nQtr = nHalf >> 1;
  if (isInverse) {
    fwdInvFactor = -1.0;
  } else {
    fwdInvFactor = 1.0;
  }

  /* For each channel */
  for (iCh = 0; iCh < nChans; iCh = iCh + 1) {
    /* Perform butterflies for the first stage, where no multiply is
required. */
    for (ix = 0; ix < fftLen - 1; ix = ix + 2) {
      i2 = ix + 1;
      twidRe = yCplx[i2].re;
      twidIm = yCplx[i2].im;
      temp[0] = twidRe;
      temp[1] = twidIm;
      yCplx[i2].re = yCplx[ix].re - temp[0];
      yCplx[i2].im = yCplx[ix].im - temp[1];
      yCplx[ix].re = yCplx[ix].re + temp[0];
      yCplx[ix].im = yCplx[ix].im + temp[1];
    }

    idelta = 2;
    k = fftLen >> 2;
    kratio = k * twiddleStep;
    while (k > 0) {
      i1 = 0;

      /* Perform the first butterfly in each remaining stage, where
no multiply is required. */
      for (ix = 0; ix < k; ix = ix + 1) {
        i2 = i1 + idelta;
        twidRe = yCplx[i2].re;
        twidIm = yCplx[i2].im;
        temp_0[0] = twidRe;
        temp_0[1] = twidIm;
        yCplx[i2].re = yCplx[i1].re - temp_0[0];
        yCplx[i2].im = yCplx[i1].im - temp_0[1];
        yCplx[i1].re = yCplx[i1].re + temp_0[0];
        yCplx[i1].im = yCplx[i1].im + temp_0[1];
        i1 = (idelta << 1) + i1;
      }

      istart = 1;
```

```
      /* Perform remaining butterflies */
      for (j = kratio; j < nHalf; j = j + kratio) {
        i1 = istart;
        twidRe = tablePtr[j];
        twidIm = tablePtr[j + nQtr] * fwdInvFactor;
        for (ix = 0; ix < k; ix = ix + 1) {
          i2 = i1 + idelta;
          tmp = yCplx[i2].re;
          tmp_0 = yCplx[i2].im;
          temp_1[0] = tmp * twidRe - tmp_0 * twidIm;
          temp_1[1] = tmp * twidIm + tmp_0 * twidRe;
          yCplx[i2].re = yCplx[i1].re - temp_1[0];
          yCplx[i2].im = yCplx[i1].im - temp_1[1];
          yCplx[i1].re = yCplx[i1].re + temp_1[0];
          yCplx[i1].im = yCplx[i1].im + temp_1[1];
          i1 = (idelta << 1) + i1;
        }

        istart = istart + 1;
      }

      idelta = idelta << 1;
      k = k >> 1;
      kratio = kratio >> 1;
    }

    /* Point to next channel */
    yCplx = &yCplx[nRows];
  }
}

void MWDSPCG_FFT_DblLen_Z_Tbl(creal_T *y, const int32_T nChans, const
int32_T
  nRows, const real_T *twiddleTable, const int32_T twiddleStep)
{
  real_T accRe;
  real_T tempOut0Re;
  real_T tempOut0Im;
  real_T tempOut1Re;
  real_T temp2Re;
  int32_T N2;
  int32_T N4;
  int32_T W4;
  int32_T yIdx;
  int32_T i;
  int32_T k;
  real_T accum;

  /* iIn-place "double-length" data recovery
     Table-based mem-optimized twiddle computation

     Used to recover linear-ordered length-N point complex FFT result
     from a linear-ordered complex length-N/2 point FFT, performed
     on N interleaved real values.
   */
  N2 = nRows >> 1;
  N4 = N2 >> 1;
  W4 = N4 * twiddleStep;
  yIdx = (nChans - 1) * nRows;
  if (nRows > 2) {
    tempOut0Re = y[N4 + yIdx].re;
```

```
  tempOut0Im = y[N4 + yIdx].im;
  y[N2 + (N4 + yIdx)].re = tempOut0Re;
  y[N2 + (N4 + yIdx)].im = tempOut0Im;
  y[N4 + yIdx].re = tempOut0Re;
  y[N4 + yIdx].im = -tempOut0Im;
}

if (nRows > 1) {
  accRe = y[yIdx].re;
  accRe = accRe - y[yIdx].im;
  y[N2 + yIdx].re = accRe;
  y[N2 + yIdx].im = 0.0;
}

accRe = y[yIdx].re;
accRe = accRe + y[yIdx].im;
y[yIdx].re = accRe;
y[yIdx].im = 0.0;
k = twiddleStep;
for (i = 1; i < N4; i = i + 1) {
  accRe = y[i + yIdx].re;
  accRe = y[(N2 - i) + yIdx].re + accRe;
  accRe = accRe / 2.0;
  temp2Re = accRe;
  accRe = y[i + yIdx].im;
  accRe = accRe - y[(N2 - i) + yIdx].im;
  accRe = accRe / 2.0;
  tempOut0Re = temp2Re;
  tempOut0Im = accRe;
  accRe = y[i + yIdx].im;
  accRe = y[(N2 - i) + yIdx].im + accRe;
  accRe = accRe / 2.0;
  tempOut1Re = accRe;
  accRe = y[(N2 - i) + yIdx].re;
  accRe = accRe - y[i + yIdx].re;
  accRe = accRe / 2.0;
  y[i + yIdx].re = tempOut1Re;
  y[i + yIdx].im = accRe;
  accRe = y[i + yIdx].re * twiddleTable[k];
  accum = accRe;
  accRe = (-twiddleTable[W4 - k]) * y[i + yIdx].im;
  accum = accum - accRe;
  tempOut1Re = accum;
  accRe = (-twiddleTable[W4 - k]) * y[i + yIdx].re;
  accum = accRe;
  accRe = y[i + yIdx].im * twiddleTable[k];
  accum = accum + accRe;
  y[i + yIdx].re = temp2Re + tempOut1Re;
  y[i + yIdx].im = tempOut0Im + accum;
  temp2Re = y[i + yIdx].re;
  accRe = -y[i + yIdx].im;
  y[(nRows - i) + yIdx].re = temp2Re;
  y[(nRows - i) + yIdx].im = accRe;
  y[N2 + (i + yIdx)].re = tempOut0Re - tempOut1Re;
  y[N2 + (i + yIdx)].im = tempOut0Im - accum;
  temp2Re = y[(i + yIdx) + N2].re;
  accRe = -y[(i + yIdx) + N2].im;
  y[(N2 - i) + yIdx].re = temp2Re;
  y[(N2 - i) + yIdx].im = accRe;
  k = k + twiddleStep;
}
```

```
}

/* Model step function for TID0 */
void schemefinal_step0(double inputtomodel, double breakerstatus)
/* Sample time: [0.00005s, 0.0s] */
{
  /* tid 0 shares data with slower tid 1 */
  if ((++schemefinal_M->Timing.RateInteraction.TID0_1) == 8)
    schemefinal_M->Timing.RateInteraction.TID0_1 = 0;
       schemefinal_B.FromWorkspace = inputtomodel; //*input //
                                                     DataValues;

  /* S-Function (sdspdsamp2): '<Root>/Downsample' */
  if ((schemefinal_M->Timing.RateInteraction.TID0_1 == 1)) {
    schemefinal_B.Downsample = schemefinal_B.FromWorkspace;
  }

  /* Update absolute time */
  schemefinal_M->Timing.clockTick0++;
}

/* Model step function for TID1 */
int schemefinal_step1(double breakerfloat)           /* Sample time:
                                                      [0.0004s, 0.0s]
*/
{
  {
    int32_T i;
    int32_T idx;
    int32_T idxW;
    real_T tmp[64];
    real_T tmp_0[64];
    real_T tmp_1;

    /* Signal Processing Blockset Buffer/Unbuffer (sdsprebuff2) -
'<Root>/Buffer' */
    {
      const byte_T *u = (const byte_T *)&schemefinal_B.Downsample;
      byte_T *y = (byte_T *)schemefinal_B.Buffer;
      byte_T *mem = (byte_T *)&schemefinal_DWork.Buffer_CircBuff[0];
      int_T uWidth = 1 * sizeof(real_T);
      int_T yWidth = 64 * sizeof(real_T);
      int_T memWidth = yWidth - uWidth;
      int_T bpeTimesNSampsAtBot;
      byte_T* bufPtr = (byte_T *)schemefinal_DWork.Buffer_IN_BUF_PTR;
      byte_T* topBuf = mem;
      byte_T* endBuf = topBuf + memWidth;

      /* copy mem to output */
      bpeTimesNSampsAtBot = (endBuf - bufPtr);
      memcpy(y, bufPtr, bpeTimesNSampsAtBot);
      memcpy(y + bpeTimesNSampsAtBot, topBuf, (memWidth-
bpeTimesNSampsAtBot));

      /* copy input to output */
      memcpy((y+memWidth), u, uWidth);

      /* copy input to output */
      memcpy(bufPtr, u, uWidth);
      bufPtr += uWidth;
```

```
      /* wrap bufPtr if it goes beyond endBuf */
      if (bufPtr >= endBuf) {
        bufPtr -= memWidth;
      }

      /* save bufPtr for next output hit */
      schemefinal_DWork.Buffer_IN_BUF_PTR = bufPtr;
    }

    /* S-Function (sdspwindow2): '<S1>/Window' */
    idx = 0;
    idxW = 0;
    for (i = 0; i < 64; i++) {
      schemefinal_B.Window_o1[idx] = schemefinal_B.Buffer[idx] *
        schemefinal_ConstP.Window_WindowSamples[idxW];
      idx++;
      idxW++;
    }

    if (!schemefinal_DWork.Window_FLAG) {
      schemefinal_DWork.Window_FLAG = true;
      idxW = 0;
      for (i = 0; i < 64; i++) {
        schemefinal_B.Window_o2[idxW] =
          schemefinal_ConstP.Window_WindowSamples[idxW];
        idxW++;
      }
    }

    /* S-Function (sdspfft2): '<S10>/FFT' */
    MWDSPCG_FFT_Interleave_R2BR_D(&schemefinal_B.Window_o1[0],
      &schemefinal_B.FFT[0], (const int32_T)1, (const int32_T)64);
    MWDSPCG_R2DIT_TBLS_Z(&schemefinal_B.FFT[0], (const int32_T)1,
(const int32_T)
                         64, (const int32_T)32, (const int32_T)0,
                         &schemefinal_ConstP.FFT_TwiddleTable[0],
(const int32_T)
                         2, (const boolean_T)false);
    MWDSPCG_FFT_DblLen_Z_Tbl(&schemefinal_B.FFT[0], (const int32_T)1,
(const
      int32_T)64, &schemefinal_ConstP.FFT_TwiddleTable[0], (const
int32_T)1);

    /* Math: '<S10>/Magnitude Squared' */
    for (idx = 0; idx < 64; idx++) {
      tmp[idx] = schemefinal_B.FFT[idx].re;
    }

    for (idx = 0; idx < 64; idx++) {
      tmp_0[idx] = schemefinal_B.FFT[idx].im;
    }

    for (idx = 0; idx < 64; idx++) {
      schemefinal_B.MagnitudeSquared[idx] = tmp[idx] * tmp[idx] +
tmp_0[idx] *
        tmp_0[idx];
    }

    /* S-Function (sfix_dot): '<S11>/Dot Product' */
    tmp_1 = 0.0;
    for (idx = 0; idx < 64; idx++) {
```

```
      tmp_1 += schemefinal_B.Window_o2[idx] *
schemefinal_B.Window_o2[idx];
    }

    schemefinal_B.DotProduct = tmp_1;

    /* Math: '<S11>/Math Function1' */
    schemefinal_B.MathFunction1 = 1.0 / schemefinal_B.DotProduct;

    /* Product: '<S1>/Product2' */
    for (idx = 0; idx < 64; idx++) {
      schemefinal_B.Product2[idx] =
schemefinal_B.MagnitudeSquared[idx] *
        schemefinal_B.MathFunction1;
    }

    /* Reshape: '<Root>/Reshape' */
    memcpy((void *)(&schemefinal_B.Reshape[0]), (void *)
           (&schemefinal_B.Product2[0]), (uint32_T)((char_T *)
            (&schemefinal_B.Product2[1]) - (char_T
*)(&schemefinal_B.Product2[0]))
           << 3U);

    /* RelationalOperator: '<S68>/Compare' incorporates:
     *  Constant: '<S68>/Constant'
     */
    schemefinal_B.Compare = (schemefinal_B.Reshape[0] <
      schemefinal_P.Constant_Value_o);

    /* Switch: '<S59>/Switch1' incorporates:
     *  Constant: '<S59>/Constant'
     *  Constant: '<S59>/Constant1'
     */
    if (schemefinal_B.Compare >= schemefinal_P.Switch1_Threshold) {
      schemefinal_B.Switch1 = schemefinal_P.Constant_Value;
    } else {
      schemefinal_B.Switch1 = schemefinal_P.Constant1_Value_n;
    }

    /* RelationalOperator: '<S67>/Compare' incorporates:
     *  Constant: '<S67>/Constant'
     */
    schemefinal_B.Compare_i = (schemefinal_B.Reshape[0] >
      schemefinal_P.Constant_Value_f);

    /* Logic: '<S59>/Logical Operator' */
    schemefinal_B.LogicalOperator = ((schemefinal_B.Compare_i != 0)
||
      (schemefinal_B.Compare != 0));

    /* Fcn: '<S59>/Compute output ' */
    schemefinal_B.Computeoutput = (schemefinal_B.Reshape[0] -
      0.06727666299001267) * 1.4317998000686814E-004 - 1.0;

    /* Switch: '<S59>/Switch' */
    if (schemefinal_B.LogicalOperator) {
      schemefinal_B.Switch = schemefinal_B.Switch1;
    } else {
      schemefinal_B.Switch = schemefinal_B.Computeoutput;
    }
```

```
    /* RelationalOperator: '<S70>/Compare' incorporates:
     *  Constant: '<S70>/Constant'
     */
    schemefinal_B.Compare_j = (schemefinal_B.Reshape[1] <
      schemefinal_P.Constant_Value_n);

    /* Switch: '<S60>/Switch1' incorporates:
     *  Constant: '<S60>/Constant'
     *  Constant: '<S60>/Constant1'
     */
    if (schemefinal_B.Compare_j >= schemefinal_P.Switch1_Threshold_d)
{
      schemefinal_B.Switch1_l = schemefinal_P.Constant_Value_p;
    } else {
      schemefinal_B.Switch1_l = schemefinal_P.Constant1_Value_c;
    }

    /* RelationalOperator: '<S69>/Compare' incorporates:
     *  Constant: '<S69>/Constant'
     */
    schemefinal_B.Compare_n = (schemefinal_B.Reshape[1] >
      schemefinal_P.Constant_Value_m);

    /* Logic: '<S60>/Logical Operator' */
    schemefinal_B.LogicalOperator_h = ((schemefinal_B.Compare_n != 0)
||
      (schemefinal_B.Compare_j != 0));

    /* Fcn: '<S60>/Compute output ' */
    schemefinal_B.Computeoutput_a = (schemefinal_B.Reshape[1] -
      442.29867797250364) * 5.8598693149398322E-005 - 1.0;

    /* Switch: '<S60>/Switch' */
    if (schemefinal_B.LogicalOperator_h) {
      schemefinal_B.Switch_j = schemefinal_B.Switch1_l;
    } else {
      schemefinal_B.Switch_j = schemefinal_B.Computeoutput_a;
    }

    /* RelationalOperator: '<S72>/Compare' incorporates:
     *  Constant: '<S72>/Constant'
     */
    schemefinal_B.Compare_nt = (schemefinal_B.Reshape[2] <
      schemefinal_P.Constant_Value_l);

    /* Switch: '<S61>/Switch1' incorporates:
     *  Constant: '<S61>/Constant'
     *  Constant: '<S61>/Constant1'
     */
    if (schemefinal_B.Compare_nt >=
schemefinal_P.Switch1_Threshold_m) {
      schemefinal_B.Switch1_o = schemefinal_P.Constant_Value_fr;
    } else {
      schemefinal_B.Switch1_o = schemefinal_P.Constant1_Value_l;
    }

    /* RelationalOperator: '<S71>/Compare' incorporates:
     *  Constant: '<S71>/Constant'
     */
    schemefinal_B.Compare_jy = (schemefinal_B.Reshape[2] >
      schemefinal_P.Constant_Value_k);
```

```
    /* Logic: '<S61>/Logical Operator' */
    schemefinal_B.LogicalOperator_l = ((schemefinal_B.Compare_jy !=
0) ||
      (schemefinal_B.Compare_nt != 0));

    /* Fcn: '<S61>/Compute output ' */
    schemefinal_B.Computeoutput_g = (schemefinal_B.Reshape[2] -
      190.64422461636815) * 6.3653614775107797E-005 - 1.0;

    /* Switch: '<S61>/Switch' */
    if (schemefinal_B.LogicalOperator_l) {
      schemefinal_B.Switch_n = schemefinal_B.Switch1_o;
    } else {
      schemefinal_B.Switch_n = schemefinal_B.Computeoutput_g;
    }

    /* RelationalOperator: '<S74>/Compare' incorporates:
     *  Constant: '<S74>/Constant'
     */
    schemefinal_B.Compare_a = (schemefinal_B.Reshape[3] <
      schemefinal_P.Constant_Value_j);

    /* Switch: '<S62>/Switch1' incorporates:
     *  Constant: '<S62>/Constant'
     *  Constant: '<S62>/Constant1'
     */
    if (schemefinal_B.Compare_a >= schemefinal_P.Switch1_Threshold_h)
{
      schemefinal_B.Switch1_f = schemefinal_P.Constant_Value_fv;
    } else {
      schemefinal_B.Switch1_f = schemefinal_P.Constant1_Value_h;
    }

    /* RelationalOperator: '<S73>/Compare' incorporates:
     *  Constant: '<S73>/Constant'
     */
    schemefinal_B.Compare_c = (schemefinal_B.Reshape[3] >
      schemefinal_P.Constant_Value_h);

    /* Logic: '<S62>/Logical Operator' */
    schemefinal_B.LogicalOperator_n = ((schemefinal_B.Compare_c != 0)
||
      (schemefinal_B.Compare_a != 0));

    /* Fcn: '<S62>/Compute output ' */
    schemefinal_B.Computeoutput_o = (schemefinal_B.Reshape[3] -
      13.689509516578283) * 2.4470113601769792E-004 - 1.0;

    /* Switch: '<S62>/Switch' */
    if (schemefinal_B.LogicalOperator_n) {
      schemefinal_B.Switch_p = schemefinal_B.Switch1_f;
    } else {
      schemefinal_B.Switch_p = schemefinal_B.Computeoutput_o;
    }

    /* RelationalOperator: '<S76>/Compare' incorporates:
     *  Constant: '<S76>/Constant'
     */
    schemefinal_B.Compare_p = (schemefinal_B.Reshape[4] <
      schemefinal_P.Constant_Value_e);
```

```
    /* Switch: '<S63>/Switch1' incorporates:
     *  Constant: '<S63>/Constant'
     *  Constant: '<S63>/Constant1'
     */
    if (schemefinal_B.Compare_p >= schemefinal_P.Switch1_Threshold_l)
{
      schemefinal_B.Switch1_n = schemefinal_P.Constant_Value_po;
    } else {
      schemefinal_B.Switch1_n = schemefinal_P.Constant1_Value_a;
    }

    /* RelationalOperator: '<S75>/Compare' incorporates:
     *  Constant: '<S75>/Constant'
     */
    schemefinal_B.Compare_g = (schemefinal_B.Reshape[4] >
      schemefinal_P.Constant_Value_d);

    /* Logic: '<S63>/Logical Operator' */
    schemefinal_B.LogicalOperator_b = ((schemefinal_B.Compare_g != 0)
||
      (schemefinal_B.Compare_p != 0));

    /* Fcn: '<S63>/Compute output ' */
    schemefinal_B.Computeoutput_gc = (schemefinal_B.Reshape[4] -
      55.987812793409645) * 8.4838352400706568E-004 - 1.0;

    /* Switch: '<S63>/Switch' */
    if (schemefinal_B.LogicalOperator_b) {
      schemefinal_B.Switch_l = schemefinal_B.Switch1_n;
    } else {
      schemefinal_B.Switch_l = schemefinal_B.Computeoutput_gc;
    }

    /* RelationalOperator: '<S78>/Compare' incorporates:
     *  Constant: '<S78>/Constant'
     */
    schemefinal_B.Compare_gp = (schemefinal_B.Reshape[5] <
      schemefinal_P.Constant_Value_g);

    /* Switch: '<S64>/Switch1' incorporates:
     *  Constant: '<S64>/Constant'
     *  Constant: '<S64>/Constant1'
     */
    if (schemefinal_B.Compare_gp >=
schemefinal_P.Switch1_Threshold_mz) {
      schemefinal_B.Switch1_om = schemefinal_P.Constant_Value_pu;
    } else {
      schemefinal_B.Switch1_om = schemefinal_P.Constant1_Value_lp;
    }

    /* RelationalOperator: '<S77>/Compare' incorporates:
     *  Constant: '<S77>/Constant'
     */
    schemefinal_B.Compare_f = (schemefinal_B.Reshape[5] >
      schemefinal_P.Constant_Value_fu);

    /* Logic: '<S64>/Logical Operator' */
    schemefinal_B.LogicalOperator_d = ((schemefinal_B.Compare_f != 0)
||
      (schemefinal_B.Compare_gp != 0));
```

```
    /* Fcn: '<S64>/Compute output ' */
    schemefinal_B.Computeoutput_c = (schemefinal_B.Reshape[5] -
      0.6056598625173274) * 1.4328585409722810E-003 - 1.0;

    /* Switch: '<S64>/Switch' */
    if (schemefinal_B.LogicalOperator_d) {
      schemefinal_B.Switch_jx = schemefinal_B.Switch1_om;
    } else {
      schemefinal_B.Switch_jx = schemefinal_B.Computeoutput_c;
    }

    /* RelationalOperator: '<S80>/Compare' incorporates:
     *  Constant: '<S80>/Constant'
     */
    schemefinal_B.Compare_fo = (schemefinal_B.Reshape[6] <
      schemefinal_P.Constant_Value_ka);

    /* Switch: '<S65>/Switch1' incorporates:
     *  Constant: '<S65>/Constant'
     *  Constant: '<S65>/Constant1'
     */
    if (schemefinal_B.Compare_fo >=
schemefinal_P.Switch1_Threshold_e) {
      schemefinal_B.Switch1_oq = schemefinal_P.Constant_Value_c;
    } else {
      schemefinal_B.Switch1_oq = schemefinal_P.Constant1_Value_j;
    }

    /* RelationalOperator: '<S79>/Compare' incorporates:
     *  Constant: '<S79>/Constant'
     */
    schemefinal_B.Compare_c4 = (schemefinal_B.Reshape[6] >
      schemefinal_P.Constant_Value_fc);

    /* Logic: '<S65>/Logical Operator' */
    schemefinal_B.LogicalOperator_i = ((schemefinal_B.Compare_c4 !=
0) ||
      (schemefinal_B.Compare_fo != 0));

    /* Fcn: '<S65>/Compute output ' */
    schemefinal_B.Computeoutput_f = (schemefinal_B.Reshape[6] -
      5.570996346795158) * 2.3451516882818367E-003 - 1.0;

    /* Switch: '<S65>/Switch' */
    if (schemefinal_B.LogicalOperator_i) {
      schemefinal_B.Switch_f = schemefinal_B.Switch1_oq;
    } else {
      schemefinal_B.Switch_f = schemefinal_B.Computeoutput_f;
    }

    /* RelationalOperator: '<S82>/Compare' incorporates:
     *  Constant: '<S82>/Constant'
     */
    schemefinal_B.Compare_a2 = (schemefinal_B.Reshape[7] <
      schemefinal_P.Constant_Value_ex);

    /* Switch: '<S66>/Switch1' incorporates:
     *  Constant: '<S66>/Constant'
     *  Constant: '<S66>/Constant1'
     */
```

```
    if (schemefinal_B.Compare_a2 >=
schemefinal_P.Switch1_Threshold_b) {
      schemefinal_B.Switch1_lj = schemefinal_P.Constant_Value_lj;
    } else {
      schemefinal_B.Switch1_lj = schemefinal_P.Constant1_Value_e;
    }

    /* RelationalOperator: '<S81>/Compare' incorporates:
     *  Constant: '<S81>/Constant'
     */
    schemefinal_B.Compare_a3 = (schemefinal_B.Reshape[7] >
      schemefinal_P.Constant_Value_i);

    /* Logic: '<S66>/Logical Operator' */
    schemefinal_B.LogicalOperator_bb = ((schemefinal_B.Compare_a3 !=
0) ||
      (schemefinal_B.Compare_a2 != 0));

    /* Fcn: '<S66>/Compute output ' */
    schemefinal_B.Computeoutput_k = (schemefinal_B.Reshape[7] -
      0.24231635126193563) * 3.3223455875474530E-003 - 1.0;

    /* Switch: '<S66>/Switch' */
    if (schemefinal_B.LogicalOperator_bb) {
      schemefinal_B.Switch_js = schemefinal_B.Switch1_lj;
    } else {
      schemefinal_B.Switch_js = schemefinal_B.Computeoutput_k;
    }

    /* Bias: '<S121>/Subtract min x' */
    schemefinal_B.Subtractminx[0] = schemefinal_B.Switch +
      schemefinal_P.Subtractminx_Bias[0];
    schemefinal_B.Subtractminx[1] = schemefinal_B.Switch_j +
      schemefinal_P.Subtractminx_Bias[1];
    schemefinal_B.Subtractminx[2] = schemefinal_B.Switch_n +
      schemefinal_P.Subtractminx_Bias[2];
    schemefinal_B.Subtractminx[3] = schemefinal_B.Switch_p +
      schemefinal_P.Subtractminx_Bias[3];
    schemefinal_B.Subtractminx[4] = schemefinal_B.Switch_l +
      schemefinal_P.Subtractminx_Bias[4];
    schemefinal_B.Subtractminx[5] = schemefinal_B.Switch_jx +
      schemefinal_P.Subtractminx_Bias[5];
    schemefinal_B.Subtractminx[6] = schemefinal_B.Switch_f +
      schemefinal_P.Subtractminx_Bias[6];
    schemefinal_B.Subtractminx[7] = schemefinal_B.Switch_js +
      schemefinal_P.Subtractminx_Bias[7];
    for (idx = 0; idx < 8; idx++) {
      /* Gain: '<S121>/Divide by range x' */
      schemefinal_B.Dividebyrangex[idx] =
schemefinal_P.Dividebyrangex_Gain[idx]
        * schemefinal_B.Subtractminx[idx];

      /* Gain: '<S121>/Multiply by range y' */
      schemefinal_B.Multiplybyrangey[idx] =
schemefinal_P.Multiplybyrangey_Gain *
        schemefinal_B.Dividebyrangex[idx];

      /* Bias: '<S121>/Add min y' */
      schemefinal_B.Addminy[idx] =
schemefinal_B.Multiplybyrangey[idx] +
        schemefinal_P.Addminy_Bias;
```

```
  /* Product: '<S103>/Product' incorporates:
   *  Constant: '<S89>/IW{1,1}(1,:)\''
   */
  schemefinal_B.Product[idx] = schemefinal_P.IW111_Value[idx] *
    schemefinal_B.Addminy[idx];
}

/* Sum: '<S103>/Sum' */
tmp_1 = schemefinal_B.Product[0];
for (idx = 0; idx < 7; idx++) {
  idxW = idx + 1;
  tmp_1 += schemefinal_B.Product[idxW];
}

schemefinal_B.Sum = tmp_1;

/* Product: '<S107>/Product' incorporates:
 *  Constant: '<S89>/IW{1,1}(2,:)\''
 */
for (idx = 0; idx < 8; idx++) {
  schemefinal_B.Product_p[idx] = schemefinal_P.IW112_Value[idx] *
    schemefinal_B.Addminy[idx];
}

/* Sum: '<S107>/Sum' */
tmp_1 = schemefinal_B.Product_p[0];
for (idx = 0; idx < 7; idx++) {
  idxW = idx + 1;
  tmp_1 += schemefinal_B.Product_p[idxW];
}

schemefinal_B.Sum_i = tmp_1;

/* Product: '<S108>/Product' incorporates:
 *  Constant: '<S89>/IW{1,1}(3,:)\''
 */
for (idx = 0; idx < 8; idx++) {
  schemefinal_B.Product_f[idx] = schemefinal_P.IW113_Value[idx] *
    schemefinal_B.Addminy[idx];
}

/* Sum: '<S108>/Sum' */
tmp_1 = schemefinal_B.Product_f[0];
for (idx = 0; idx < 7; idx++) {
  idxW = idx + 1;
  tmp_1 += schemefinal_B.Product_f[idxW];
}

schemefinal_B.Sum_j = tmp_1;

/* Product: '<S109>/Product' incorporates:
 *  Constant: '<S89>/IW{1,1}(4,:)\''
 */
for (idx = 0; idx < 8; idx++) {
  schemefinal_B.Product_i[idx] = schemefinal_P.IW114_Value[idx] *
    schemefinal_B.Addminy[idx];
}

/* Sum: '<S109>/Sum' */
tmp_1 = schemefinal_B.Product_i[0];
```

```
for (idx = 0; idx < 7; idx++) {
  idxW = idx + 1;
  tmp_1 += schemefinal_B.Product_i[idxW];
}

schemefinal_B.Sum_iu = tmp_1;

/* Product: '<S110>/Product' incorporates:
 *  Constant: '<S89>/IW{1,1}(5,:)\''
 */
for (idx = 0; idx < 8; idx++) {
  schemefinal_B.Product_g[idx] = schemefinal_P.IW115_Value[idx] *
    schemefinal_B.Addminy[idx];
}

/* Sum: '<S110>/Sum' */
tmp_1 = schemefinal_B.Product_g[0];
for (idx = 0; idx < 7; idx++) {
  idxW = idx + 1;
  tmp_1 += schemefinal_B.Product_g[idxW];
}

schemefinal_B.Sum_f = tmp_1;

/* Product: '<S111>/Product' incorporates:
 *  Constant: '<S89>/IW{1,1}(6,:)\''
 */
for (idx = 0; idx < 8; idx++) {
  schemefinal_B.Product_l[idx] = schemefinal_P.IW116_Value[idx] *
    schemefinal_B.Addminy[idx];
}

/* Sum: '<S111>/Sum' */
tmp_1 = schemefinal_B.Product_l[0];
for (idx = 0; idx < 7; idx++) {
  idxW = idx + 1;
  tmp_1 += schemefinal_B.Product_l[idxW];
}

schemefinal_B.Sum_a = tmp_1;

/* Product: '<S112>/Product' incorporates:
 *  Constant: '<S89>/IW{1,1}(7,:)\''
 */
for (idx = 0; idx < 8; idx++) {
  schemefinal_B.Product_e[idx] = schemefinal_P.IW117_Value[idx] *
    schemefinal_B.Addminy[idx];
}

/* Sum: '<S112>/Sum' */
tmp_1 = schemefinal_B.Product_e[0];
for (idx = 0; idx < 7; idx++) {
  idxW = idx + 1;
  tmp_1 += schemefinal_B.Product_e[idxW];
}

schemefinal_B.Sum_o = tmp_1;

/* Product: '<S113>/Product' incorporates:
 *  Constant: '<S89>/IW{1,1}(8,:)\''
 */
```

```
    for (idx = 0; idx < 8; idx++) {
      schemefinal_B.Product_gr[idx] = schemefinal_P.IW118_Value[idx]
*
        schemefinal_B.Addminy[idx];
    }

    /* Sum: '<S113>/Sum' */
    tmp_1 = schemefinal_B.Product_gr[0];
    for (idx = 0; idx < 7; idx++) {
      idxW = idx + 1;
      tmp_1 += schemefinal_B.Product_gr[idxW];
    }

    schemefinal_B.Sum_g = tmp_1;

    /* Product: '<S114>/Product' incorporates:
     *  Constant: '<S89>/IW{1,1}(9,:)\''
     */
    for (idx = 0; idx < 8; idx++) {
      schemefinal_B.Product_f4[idx] = schemefinal_P.IW119_Value[idx]
*
        schemefinal_B.Addminy[idx];
    }

    /* Sum: '<S114>/Sum' */
    tmp_1 = schemefinal_B.Product_f4[0];
    for (idx = 0; idx < 7; idx++) {
      idxW = idx + 1;
      tmp_1 += schemefinal_B.Product_f4[idxW];
    }

    schemefinal_B.Sum_d = tmp_1;

    /* Product: '<S104>/Product' incorporates:
     *  Constant: '<S89>/IW{1,1}(10,:)\''
     */
    for (idx = 0; idx < 8; idx++) {
      schemefinal_B.Product_ep[idx] = schemefinal_P.IW1110_Value[idx]
*
        schemefinal_B.Addminy[idx];
    }

    /* Sum: '<S104>/Sum' */
    tmp_1 = schemefinal_B.Product_ep[0];
    for (idx = 0; idx < 7; idx++) {
      idxW = idx + 1;
      tmp_1 += schemefinal_B.Product_ep[idxW];
    }

    schemefinal_B.Sum_l = tmp_1;

    /* Product: '<S105>/Product' incorporates:
     *  Constant: '<S89>/IW{1,1}(11,:)\''
     */
    for (idx = 0; idx < 8; idx++) {
      schemefinal_B.Product_n[idx] = schemefinal_P.IW1111_Value[idx]
*
        schemefinal_B.Addminy[idx];
    }

    /* Sum: '<S105>/Sum' */
```

```
    tmp_1 = schemefinal_B.Product_n[0];
    for (idx = 0; idx < 7; idx++) {
      idxW = idx + 1;
      tmp_1 += schemefinal_B.Product_n[idxW];
    }

    schemefinal_B.Sum_jn = tmp_1;

    /* Product: '<S106>/Product' incorporates:
     *  Constant: '<S89>/IW{1,1}(12,:)\''
     */
    for (idx = 0; idx < 8; idx++) {
      schemefinal_B.Product_j[idx] = schemefinal_P.IW1112_Value[idx]
*
        schemefinal_B.Addminy[idx];
    }

    /* Sum: '<S106>/Sum' */
    tmp_1 = schemefinal_B.Product_j[0];
    for (idx = 0; idx < 7; idx++) {
      idxW = idx + 1;
      tmp_1 += schemefinal_B.Product_j[idxW];
    }

    schemefinal_B.Sum_oz = tmp_1;

    /* Sum: '<S84>/netsum' incorporates:
     *  Constant: '<S84>/b{1}'
     */
    schemefinal_B.netsum[0] = schemefinal_B.Sum +
schemefinal_P.b1_Value[0];
    schemefinal_B.netsum[1] = schemefinal_B.Sum_i +
schemefinal_P.b1_Value[1];
    schemefinal_B.netsum[2] = schemefinal_B.Sum_j +
schemefinal_P.b1_Value[2];
    schemefinal_B.netsum[3] = schemefinal_B.Sum_iu +
schemefinal_P.b1_Value[3];
    schemefinal_B.netsum[4] = schemefinal_B.Sum_f +
schemefinal_P.b1_Value[4];
    schemefinal_B.netsum[5] = schemefinal_B.Sum_a +
schemefinal_P.b1_Value[5];
    schemefinal_B.netsum[6] = schemefinal_B.Sum_o +
schemefinal_P.b1_Value[6];
    schemefinal_B.netsum[7] = schemefinal_B.Sum_g +
schemefinal_P.b1_Value[7];
    schemefinal_B.netsum[8] = schemefinal_B.Sum_d +
schemefinal_P.b1_Value[8];
    schemefinal_B.netsum[9] = schemefinal_B.Sum_l +
schemefinal_P.b1_Value[9];
    schemefinal_B.netsum[10] = schemefinal_B.Sum_jn +
schemefinal_P.b1_Value[10];
    schemefinal_B.netsum[11] = schemefinal_B.Sum_oz +
schemefinal_P.b1_Value[11];

    /* Product: '<S119>/Product' incorporates:
     *  Constant: '<S116>/IW{2,1}(1,:)\''
     */
    for (idx = 0; idx < 12; idx++) {
      schemefinal_B.Product_n3[idx] = schemefinal_P.IW211_Value[idx]
*
        schemefinal_B.netsum[idx];
```

```
    }

    /* Sum: '<S119>/Sum' */
    tmp_1 = schemefinal_B.Product_n3[0];
    for (idx = 0; idx < 11; idx++) {
      idxW = idx + 1;
      tmp_1 += schemefinal_B.Product_n3[idxW];
    }

    schemefinal_B.Sum_p = tmp_1;

    /* Sum: '<S85>/netsum' incorporates:
     *  Constant: '<S85>/b{2}'
     */
    schemefinal_B.netsum_k = schemefinal_B.Sum_p +
schemefinal_P.b2_Value;

    /* Gain: '<S117>/Gain' */
    schemefinal_B.Gain = schemefinal_P.Gain_Gain *
schemefinal_B.netsum_k;

    /* ElementaryMath Block: '<S117>/Exp'
     */
    schemefinal_B.Exp = exp(schemefinal_B.Gain);

    /* Sum: '<S117>/Sum' incorporates:
     *  Constant: '<S117>/one'
     */
    schemefinal_B.Sum_m = schemefinal_B.Exp +
schemefinal_P.one_Value;

    /* ElementaryMath Block: '<S117>/Reciprocal'
     */
    schemefinal_B.Reciprocal = 1.0/(schemefinal_B.Sum_m);

    /* Bias: '<S123>/Subtract min y' */
    schemefinal_B.Subtractminy = schemefinal_B.Reciprocal +
      schemefinal_P.Subtractminy_Bias;

    /* Gain: '<S123>/Divide by range y' */
    schemefinal_B.Dividebyrangey = schemefinal_P.Dividebyrangey_Gain
*
      schemefinal_B.Subtractminy;

    /* Gain: '<S123>/Multiply by range x' */
    schemefinal_B.Multiplybyrangex =
schemefinal_P.Multiplybyrangex_Gain *
      schemefinal_B.Dividebyrangey;

    /* Bias: '<S123>/Add min x' */
    schemefinal_B.Addminx = schemefinal_B.Multiplybyrangex +
      schemefinal_P.Addminx_Bias;

    /* Sum: '<S4>/Sum1' incorporates:
     *  Constant: '<S4>/Constant1'
     */
    schemefinal_B.Sum1 = schemefinal_B.Addminx -
schemefinal_P.Constant1_Value_d;

    /* Gain: '<S4>/Gain' */
```

```
    schemefinal_B.Gain_b = schemefinal_P.Gain_Gain_n *
schemefinal_B.Sum1;

    /* Sum: '<S4>/Sum2' incorporates:
     *  Constant: '<S4>/Constant2'
     */
    schemefinal_B.Sum2 = schemefinal_B.Gain_b -
schemefinal_P.Constant2_Value;

    /* RelationalOperator: '<S56>/Compare' */
    schemefinal_B.Compare_je = (schemefinal_B.Sum2 >= 0.0);

    /* RelationalOperator: '<S57>/Compare' */
    schemefinal_B.Compare_iv = (schemefinal_B.Sum2 <= 0.0);


    /* DataTypeConversion: '<S58>/Data Type Conversion' */
    schemefinal_B.DataTypeConversion = (schemefinal_B.FromWorkspace_d
!= 0.0);

    /* S-Function (sdspdelay): '<S58>/Delay' */
    schemefinal_B.Delay = schemefinal_DWork.Delay_IC_BUFF;

    /* Logic: '<S58>/Logical Operator' */
    schemefinal_B.LogicalOperator_dz = !schemefinal_B.Delay;

    /* Logic: '<S58>/Logical Operator1' */
    schemefinal_B.LogicalOperator1 =
(schemefinal_B.DataTypeConversion &&
      schemefinal_B.LogicalOperator_dz);

    /* Signal Processing Blockset Boolean Converter (sdspboolconv) -
'<S58>/Logical Converter' - Output */
    {
      const boolean_T *u1 = &schemefinal_B.LogicalOperator1;
      real_T *y = &schemefinal_B.LogicalConverter;
      int_T arraySize = 1;
      while (arraySize--) {
        *y++ = (real_T)((int)(*u1++));
      }
    }

    /* Logic: '<S4>/Logical Operator' */
    schemefinal_B.LogicalOperator_a = ((schemefinal_B.Compare_iv !=
0) ||
      (schemefinal_B.LogicalConverter != 0.0));

    /* S-Function (sdspcount2): '<S4>/Non zero =  increment counter'
*/
    schemefinal_B.Nonzeroincrementcounter_o2 = 0.0;
    if (schemefinal_B.LogicalOperator_a) {
      schemefinal_DWork.Nonzeroincrementcounter_Count =
        schemefinal_P.Nonzeroincrementcounter_Initial;
    }

    if (schemefinal_B.Compare_je != 0) {
      if (schemefinal_DWork.Nonzeroincrementcounter_Count <
          schemefinal_ConstP.Nonzeroincrementcounter_Ma) {
        schemefinal_DWork.Nonzeroincrementcounter_Count =
(uint8_T)(uint32_T)
          (schemefinal_DWork.Nonzeroincrementcounter_Count + 1);
```

```
      } else {
        schemefinal_DWork.Nonzeroincrementcounter_Count = 0U;
      }
    }

    schemefinal_B.Nonzeroincrementcounter_o1 = (real_T)
      schemefinal_DWork.Nonzeroincrementcounter_Count;
    if (schemefinal_DWork.Nonzeroincrementcounter_Count ==
        schemefinal_P.Nonzeroincrementcounter_HitValu) {
      schemefinal_B.Nonzeroincrementcounter_o2 = 1.0;
    }

    /* Logic: '<Root>/AND' */
    schemefinal_B.AND = ((schemefinal_B.Nonzeroincrementcounter_o2 !=
0.0) &&
                         (breakerfloat != 0.0));

    /* Outputs for enable SubSystem: '<S7>/D Latch' incorporates:
     *  EnablePort: '<S83>/C'
     */
    if (schemefinal_B.AND) {
      if (schemefinal_DWork.DLatch_MODE == SUBSYS_DISABLED) {
        schemefinal_DWork.DLatch_MODE = SUBSYS_ENABLED;
      }
    } else {
      if (schemefinal_DWork.DLatch_MODE == SUBSYS_ENABLED) {
        schemefinal_DWork.DLatch_MODE = SUBSYS_DISABLED;
      }
    }

    if (schemefinal_DWork.DLatch_MODE == SUBSYS_ENABLED) {
      /* Inport: '<S83>/D' incorporates:
       *  Constant: '<Root>/Constant1'
       */
      schemefinal_B.D = schemefinal_P.Constant1_Value;

      /* Logic: '<S83>/Logic' */
      schemefinal_B.Logic = (real_T)!(schemefinal_B.D != 0.0);
    }

    /* end of Outputs for SubSystem: '<S7>/D Latch' */

    /* Bias: '<S50>/Subtract min x' */
    schemefinal_B.Subtractminx_h[0] = schemefinal_B.Switch +
      schemefinal_P.Subtractminx_Bias_f[0];
    schemefinal_B.Subtractminx_h[1] = schemefinal_B.Switch_j +
      schemefinal_P.Subtractminx_Bias_f[1];
    schemefinal_B.Subtractminx_h[2] = schemefinal_B.Switch_n +
      schemefinal_P.Subtractminx_Bias_f[2];
    schemefinal_B.Subtractminx_h[3] = schemefinal_B.Switch_p +
      schemefinal_P.Subtractminx_Bias_f[3];
    schemefinal_B.Subtractminx_h[4] = schemefinal_B.Switch_l +
      schemefinal_P.Subtractminx_Bias_f[4];
    schemefinal_B.Subtractminx_h[5] = schemefinal_B.Switch_jx +
      schemefinal_P.Subtractminx_Bias_f[5];
    schemefinal_B.Subtractminx_h[6] = schemefinal_B.Switch_f +
      schemefinal_P.Subtractminx_Bias_f[6];
    schemefinal_B.Subtractminx_h[7] = schemefinal_B.Switch_js +
      schemefinal_P.Subtractminx_Bias_f[7];
    for (idx = 0; idx < 8; idx++) {
      /* Gain: '<S50>/Divide by range x' */
```

```
      schemefinal_B.Dividebyrangex_a[idx] =
        schemefinal_P.Dividebyrangex_Gain_p[idx] *
        schemefinal_B.Subtractminx_h[idx];

      /* Gain: '<S50>/Multiply by range y' */
      schemefinal_B.Multiplybyrangey_p[idx] =
        schemefinal_P.Multiplybyrangey_Gain_k *
        schemefinal_B.Dividebyrangex_a[idx];

      /* Bias: '<S50>/Add min y' */
      schemefinal_B.Addminy_e[idx] =
schemefinal_B.Multiplybyrangey_p[idx] +
        schemefinal_P.Addminy_Bias_n;

      /* Product: '<S32>/Product' incorporates:
       *  Constant: '<S18>/IW{1,1}(1,:)\''
       */
      schemefinal_B.Product_lj[idx] =
schemefinal_P.IW111_Value_p[idx] *
        schemefinal_B.Addminy_e[idx];
    }

    /* Sum: '<S32>/Sum' */
    tmp_1 = schemefinal_B.Product_lj[0];
    for (idx = 0; idx < 7; idx++) {
      idxW = idx + 1;
      tmp_1 += schemefinal_B.Product_lj[idxW];
    }

    schemefinal_B.Sum_l4 = tmp_1;

    /* Product: '<S36>/Product' incorporates:
     *  Constant: '<S18>/IW{1,1}(2,:)\''
     */
    for (idx = 0; idx < 8; idx++) {
      schemefinal_B.Product_eo[idx] =
schemefinal_P.IW112_Value_n[idx] *
        schemefinal_B.Addminy_e[idx];
    }

    /* Sum: '<S36>/Sum' */
    tmp_1 = schemefinal_B.Product_eo[0];
    for (idx = 0; idx < 7; idx++) {
      idxW = idx + 1;
      tmp_1 += schemefinal_B.Product_eo[idxW];
    }

    schemefinal_B.Sum_e = tmp_1;

    /* Product: '<S37>/Product' incorporates:
     *  Constant: '<S18>/IW{1,1}(3,:)\''
     */
    for (idx = 0; idx < 8; idx++) {
      schemefinal_B.Product_k[idx] = schemefinal_P.IW113_Value_k[idx]
*
        schemefinal_B.Addminy_e[idx];
    }

    /* Sum: '<S37>/Sum' */
    tmp_1 = schemefinal_B.Product_k[0];
    for (idx = 0; idx < 7; idx++) {
```

```
      idxW = idx + 1;
      tmp_1 += schemefinal_B.Product_k[idxW];
    }

    schemefinal_B.Sum_gn = tmp_1;

    /* Product: '<S38>/Product' incorporates:
     *  Constant: '<S18>/IW{1,1}(4,:)\''
     */
    for (idx = 0; idx < 8; idx++) {
      schemefinal_B.Product_gl[idx] =
schemefinal_P.IW114_Value_h[idx] *
        schemefinal_B.Addminy_e[idx];
    }

    /* Sum: '<S38>/Sum' */
    tmp_1 = schemefinal_B.Product_gl[0];
    for (idx = 0; idx < 7; idx++) {
      idxW = idx + 1;
      tmp_1 += schemefinal_B.Product_gl[idxW];
    }

    schemefinal_B.Sum_k = tmp_1;

    /* Product: '<S39>/Product' incorporates:
     *  Constant: '<S18>/IW{1,1}(5,:)\''
     */
    for (idx = 0; idx < 8; idx++) {
      schemefinal_B.Product_go[idx] =
schemefinal_P.IW115_Value_o[idx] *
        schemefinal_B.Addminy_e[idx];
    }

    /* Sum: '<S39>/Sum' */
    tmp_1 = schemefinal_B.Product_go[0];
    for (idx = 0; idx < 7; idx++) {
      idxW = idx + 1;
      tmp_1 += schemefinal_B.Product_go[idxW];
    }

    schemefinal_B.Sum_px = tmp_1;

    /* Product: '<S40>/Product' incorporates:
     *  Constant: '<S18>/IW{1,1}(6,:)\''
     */
    for (idx = 0; idx < 8; idx++) {
      schemefinal_B.Product_l5[idx] =
schemefinal_P.IW116_Value_b[idx] *
        schemefinal_B.Addminy_e[idx];
    }

    /* Sum: '<S40>/Sum' */
    tmp_1 = schemefinal_B.Product_l5[0];
    for (idx = 0; idx < 7; idx++) {
      idxW = idx + 1;
      tmp_1 += schemefinal_B.Product_l5[idxW];
    }

    schemefinal_B.Sum_ew = tmp_1;

    /* Product: '<S41>/Product' incorporates:
```

```
     *  Constant: '<S18>/IW{1,1}(7,:)\''
     */
    for (idx = 0; idx < 8; idx++) {
      schemefinal_B.Product_b[idx] = schemefinal_P.IW117_Value_k[idx]
*
        schemefinal_B.Addminy_e[idx];
    }

    /* Sum: '<S41>/Sum' */
    tmp_1 = schemefinal_B.Product_b[0];
    for (idx = 0; idx < 7; idx++) {
      idxW = idx + 1;
      tmp_1 += schemefinal_B.Product_b[idxW];
    }

    schemefinal_B.Sum_jz = tmp_1;

    /* Product: '<S42>/Product' incorporates:
     *  Constant: '<S18>/IW{1,1}(8,:)\''
     */
    for (idx = 0; idx < 8; idx++) {
      schemefinal_B.Product_kd[idx] =
schemefinal_P.IW118_Value_o[idx] *
        schemefinal_B.Addminy_e[idx];
    }

    /* Sum: '<S42>/Sum' */
    tmp_1 = schemefinal_B.Product_kd[0];
    for (idx = 0; idx < 7; idx++) {
      idxW = idx + 1;
      tmp_1 += schemefinal_B.Product_kd[idxW];
    }

    schemefinal_B.Sum_ar = tmp_1;

    /* Product: '<S43>/Product' incorporates:
     *  Constant: '<S18>/IW{1,1}(9,:)\''
     */
    for (idx = 0; idx < 8; idx++) {
      schemefinal_B.Product_ig[idx] =
schemefinal_P.IW119_Value_i[idx] *
        schemefinal_B.Addminy_e[idx];
    }

    /* Sum: '<S43>/Sum' */
    tmp_1 = schemefinal_B.Product_ig[0];
    for (idx = 0; idx < 7; idx++) {
      idxW = idx + 1;
      tmp_1 += schemefinal_B.Product_ig[idxW];
    }

    schemefinal_B.Sum_op = tmp_1;

    /* Product: '<S33>/Product' incorporates:
     *  Constant: '<S18>/IW{1,1}(10,:)\''
     */
    for (idx = 0; idx < 8; idx++) {
      schemefinal_B.Product_lv[idx] =
schemefinal_P.IW1110_Value_e[idx] *
        schemefinal_B.Addminy_e[idx];
    }
```

```
    /* Sum: '<S33>/Sum' */
    tmp_1 = schemefinal_B.Product_lv[0];
    for (idx = 0; idx < 7; idx++) {
      idxW = idx + 1;
      tmp_1 += schemefinal_B.Product_lv[idxW];
    }

    schemefinal_B.Sum_fm = tmp_1;

    /* Product: '<S34>/Product' incorporates:
     *  Constant: '<S18>/IW{1,1}(11,:)\''
     */
    for (idx = 0; idx < 8; idx++) {
      schemefinal_B.Product_d[idx] =
schemefinal_P.IW1111_Value_g[idx] *
        schemefinal_B.Addminy_e[idx];
    }

    /* Sum: '<S34>/Sum' */
    tmp_1 = schemefinal_B.Product_d[0];
    for (idx = 0; idx < 7; idx++) {
      idxW = idx + 1;
      tmp_1 += schemefinal_B.Product_d[idxW];
    }

    schemefinal_B.Sum_ix = tmp_1;

    /* Product: '<S35>/Product' incorporates:
     *  Constant: '<S18>/IW{1,1}(12,:)\''
     */
    for (idx = 0; idx < 8; idx++) {
      schemefinal_B.Product_jd[idx] =
schemefinal_P.IW1112_Value_h[idx] *
        schemefinal_B.Addminy_e[idx];
    }

    /* Sum: '<S35>/Sum' */
    tmp_1 = schemefinal_B.Product_jd[0];
    for (idx = 0; idx < 7; idx++) {
      idxW = idx + 1;
      tmp_1 += schemefinal_B.Product_jd[idxW];
    }

    schemefinal_B.Sum_jv = tmp_1;

    /* Sum: '<S13>/netsum' incorporates:
     *  Constant: '<S13>/b{1}'
     */
    schemefinal_B.netsum_p[0] = schemefinal_B.Sum_l4 +
schemefinal_P.b1_Value_i
      [0];
    schemefinal_B.netsum_p[1] = schemefinal_B.Sum_e +
schemefinal_P.b1_Value_i[1];
    schemefinal_B.netsum_p[2] = schemefinal_B.Sum_gn +
schemefinal_P.b1_Value_i
      [2];
    schemefinal_B.netsum_p[3] = schemefinal_B.Sum_k +
schemefinal_P.b1_Value_i[3];
    schemefinal_B.netsum_p[4] = schemefinal_B.Sum_px +
schemefinal_P.b1_Value_i
```

```
      [4];
    schemefinal_B.netsum_p[5] = schemefinal_B.Sum_ew +
schemefinal_P.b1_Value_i
      [5];
    schemefinal_B.netsum_p[6] = schemefinal_B.Sum_jz +
schemefinal_P.b1_Value_i
      [6];
    schemefinal_B.netsum_p[7] = schemefinal_B.Sum_ar +
schemefinal_P.b1_Value_i
      [7];
    schemefinal_B.netsum_p[8] = schemefinal_B.Sum_op +
schemefinal_P.b1_Value_i
      [8];
    schemefinal_B.netsum_p[9] = schemefinal_B.Sum_fm +
schemefinal_P.b1_Value_i
      [9];
    schemefinal_B.netsum_p[10] = schemefinal_B.Sum_ix +
      schemefinal_P.b1_Value_i[10];
    schemefinal_B.netsum_p[11] = schemefinal_B.Sum_jv +
      schemefinal_P.b1_Value_i[11];

    /* Product: '<S48>/Product' incorporates:
     *  Constant: '<S45>/IW{2,1}(1,:)\''
     */
    for (idx = 0; idx < 12; idx++) {
      schemefinal_B.Product_jc[idx] =
schemefinal_P.IW211_Value_a[idx] *
        schemefinal_B.netsum_p[idx];
    }

    /* Sum: '<S48>/Sum' */
    tmp_1 = schemefinal_B.Product_jc[0];
    for (idx = 0; idx < 11; idx++) {
      idxW = idx + 1;
      tmp_1 += schemefinal_B.Product_jc[idxW];
    }

    schemefinal_B.Sum_fw = tmp_1;

    /* Sum: '<S14>/netsum' incorporates:
     *  Constant: '<S14>/b{2}'
     */
    schemefinal_B.netsum_px = schemefinal_B.Sum_fw +
schemefinal_P.b2_Value_a;

    /* Gain: '<S46>/Gain' */
    schemefinal_B.Gain_l = schemefinal_P.Gain_Gain_p *
schemefinal_B.netsum_px;

    /* ElementaryMath Block: '<S46>/Exp'
     */
    schemefinal_B.Exp_n = exp(schemefinal_B.Gain_l);

    /* Sum: '<S46>/Sum' incorporates:
     *  Constant: '<S46>/one'
     */
    schemefinal_B.Sum_lo = schemefinal_B.Exp_n +
schemefinal_P.one_Value_j;

    /* ElementaryMath Block: '<S46>/Reciprocal'
     */
```

```
    schemefinal_B.Reciprocal_e = 1.0/(schemefinal_B.Sum_lo);

    /* Bias: '<S52>/Subtract min y' */
    schemefinal_B.Subtractminy_f = schemefinal_B.Reciprocal_e +
      schemefinal_P.Subtractminy_Bias_p;

    /* Gain: '<S52>/Divide by range y' */
    schemefinal_B.Dividebyrangey_o =
schemefinal_P.Dividebyrangey_Gain_c *
      schemefinal_B.Subtractminy_f;

    /* Gain: '<S52>/Multiply by range x' */
    schemefinal_B.Multiplybyrangex_n =
schemefinal_P.Multiplybyrangex_Gain_o *
      schemefinal_B.Dividebyrangey_o;

    /* Bias: '<S52>/Add min x' */
    schemefinal_B.Addminx_p = schemefinal_B.Multiplybyrangex_n +
      schemefinal_P.Addminx_Bias_l;

       /* Sum: '<S3>/Sum1' incorporates:
     *  Constant: '<S3>/Constant1'
     */
    schemefinal_B.Sum1_m = schemefinal_B.Addminx_p -
      schemefinal_P.Constant1_Value_nh;

    /* RelationalOperator: '<S53>/Compare' */
    schemefinal_B.Compare_l = (schemefinal_B.Sum1_m >= 0.0);

    /* RelationalOperator: '<S54>/Compare' */
    schemefinal_B.Compare_k = (schemefinal_B.Sum1_m <= 0.0);


    /* DataTypeConversion: '<S55>/Data Type Conversion' */
    schemefinal_B.DataTypeConversion_g = (breakerfloat != 0.0);;

    /* S-Function (sdspdelay): '<S55>/Delay' */
    schemefinal_B.Delay_d = schemefinal_DWork.Delay_IC_BUFF_a;

    /* Logic: '<S55>/Logical Operator' */
    schemefinal_B.LogicalOperator_lz = !schemefinal_B.Delay_d;

    /* Logic: '<S55>/Logical Operator1' */
    schemefinal_B.LogicalOperator1_i =
(schemefinal_B.DataTypeConversion_g &&
      schemefinal_B.LogicalOperator_lz);

    /* Signal Processing Blockset Boolean Converter (sdspboolconv) -
'<S55>/Logical Converter' - Output */
    {
      const boolean_T *u1 = &schemefinal_B.LogicalOperator1_i;
      real_T *y = &schemefinal_B.LogicalConverter_g;
      int_T arraySize = 1;
      while (arraySize--) {
        *y++ = (real_T)((int)(*u1++));
      }
    }

    /* Logic: '<S3>/Logical Operator' */
    schemefinal_B.LogicalOperator_n0 = ((schemefinal_B.Compare_k !=
0) ||
```

```
      (schemefinal_B.LogicalConverter_g != 0.0));

    /* S-Function (sdspcount2): '<S3>/non zero =  increment counte'
*/
    schemefinal_B.nonzeroincrementcounte_o2 = 0.0;
    if (schemefinal_B.LogicalOperator_n0) {
      schemefinal_DWork.nonzeroincrementcounte_Count =
        schemefinal_P.nonzeroincrementcounte_InitialC;
    }

    if (schemefinal_B.Compare_l != 0) {
      if (schemefinal_DWork.nonzeroincrementcounte_Count <
          schemefinal_ConstP.nonzeroincrementcounte_Max) {
        schemefinal_DWork.nonzeroincrementcounte_Count =
(uint8_T)(uint32_T)
          (schemefinal_DWork.nonzeroincrementcounte_Count + 1);
      } else {
        schemefinal_DWork.nonzeroincrementcounte_Count = 0U;
      }
    }

    schemefinal_B.nonzeroincrementcounte_o1 = (real_T)
      schemefinal_DWork.nonzeroincrementcounte_Count;
    if (schemefinal_DWork.nonzeroincrementcounte_Count ==
        schemefinal_P.nonzeroincrementcounte_HitValue) {
      schemefinal_B.nonzeroincrementcounte_o2 = 1.0;
    }

    /* Logic: '<Root>/AND2 ' */
    int littlemodelout = ((schemefinal_B.D != 0.0) &&
                          (schemefinal_B.nonzeroincrementcounte_o2 !=
0.0));

    /* Update for S-Function (sdspdelay): '<S58>/Delay' */
    schemefinal_DWork.Delay_IC_BUFF =
schemefinal_B.DataTypeConversion;

    /* Update for S-Function (sdspdelay): '<S55>/Delay' */
    schemefinal_DWork.Delay_IC_BUFF_a =
schemefinal_B.DataTypeConversion_g;

      //write output to array


       schemefinal_B.Addminx_p;


      return littlemodelout;

      }


  /* Update absolute time */
  schemefinal_M->Timing.clockTick1++;
}

/* Model initialize function */
void schemefinal_initialize(boolean_T firstTime)
{
  (void)firstTime;
```

```
  /* Registration code */

  /* initialize non-finites */
  rt_InitInfAndNaN(sizeof(real_T));

  /* initialize real-time model */
  (void) memset((void *)schemefinal_M,0,
                sizeof(RT_MODEL_schemefinal));

  /* block I/O */
  (void) memset(((void *) &schemefinal_B),0,
                sizeof(BlockIO_schemefinal));

  /* states (dwork) */
  (void) memset((void *)&schemefinal_DWork, 0,
                sizeof(D_Work_schemefinal));

  /* Initialize DataMapInfo substructure containing ModelMap for C
API */
  schemefinal_InitializeDataMapInfo(schemefinal_M);


  /* Copy ICs into circular buffer */
  {
    const int_T bufLenBytes = 63 * sizeof(real_T);
    byte_T *circBufPtr = (byte_T
*)&schemefinal_DWork.Buffer_CircBuff[0];
    const byte_T *icPtr = (const byte_T
*)&schemefinal_ConstP.pooled1;
    int_T i = 1;
    while (i-- > 0) {
      MWDSP_CopyScalarICs(circBufPtr, icPtr, 63, sizeof(real_T));
      circBufPtr += bufLenBytes;
    }
  }

  *&schemefinal_DWork.Buffer_IN_BUF_PTR = (void *)( (byte_T *)
    &schemefinal_DWork.Buffer_CircBuff[0] );

  /* InitializeConditions for S-Function (sdspwindow2): '<S1>/Window'
*/
  schemefinal_DWork.Window_FLAG = false;

  /* InitializeConditions for S-Function (sdspdelay): '<S58>/Delay'
*/
  schemefinal_DWork.Delay_IC_BUFF = schemefinal_ConstP.pooled3;

  /* InitializeConditions for S-Function (sdspcount2): '<S4>/Non zero
=  increment counter' */
  schemefinal_DWork.Nonzeroincrementcounter_Count =
    schemefinal_P.Nonzeroincrementcounter_Initial;

  /* InitializeConditions for S-Function (sdspdelay): '<S55>/Delay'
*/
  schemefinal_DWork.Delay_IC_BUFF_a = schemefinal_ConstP.pooled3;

  /* InitializeConditions for S-Function (sdspcount2): '<S3>/non zero
=  increment counte' */
  schemefinal_DWork.nonzeroincrementcounte_Count =
    schemefinal_P.nonzeroincrementcounte_InitialC;
}
```

```
/* Model terminate function */
void schemefinal_terminate(void)
{
  /* (no terminate code required) */
}

/* File trailer for Real-Time Workshop generated code.
 *
 * [EOF]
 */
```

3) The data for the algorithm is stored in structures within `schemefinal_data.cpp`. These give the ANN weights, as well as other simulink block parameters

```
/*
 * File: schemefinal_data.cpp
 *
 * Real-Time Workshop code generated for Simulink model schemefinal.
 *
 * Model version                        : 1.79
 * Real-Time Workshop file version      : 7.2  (R2008b)  04-Aug-2008
 * Real-Time Workshop file generated on : Thu Jan 13 15:14:25 2011
 * TLC version                          : 7.2 (Aug  5 2008)
 * C/C++ source code generated on       : Thu Jan 13 15:14:27 2011
 */

#include "schemefinal.h"
#include "schemefinal_private.h"

/* Block parameters (auto storage) */
Parameters_schemefinal schemefinal_P = {
  1.0,                                 /* Constant1_Value :
'<Root>/Constant1'
                                        */
  0.5,                                 /* Constant2_Value :
'<S4>/Constant2'
                                        */
  1.0,                                 /* Constant1_Value_d :
'<S4>/Constant1'
                                        */

  /*  IW211_Value : '<S116>/IW{2,1}(1,:)''
   */
  { -2.8274178769075590E+001, 5.2130362093088034E-001, -
2.1444868781774437E-001,
    -1.3490954060794214E+000, 1.3993344053233376E+000,
2.3672486121021080E+001,
    6.5221908412841092E+000, -2.1152991037783228E-001,
4.0181532056091484E+001,
    4.7013182896155165E-001, -8.1215991089172446E-001, -
1.0539831084946492E+000
  },

  /*  IW111_Value : '<S89>/IW{1,1}(1,:)''
   */
  { -5.7304449422208137E-001, -2.3704384127416014E-001,
6.5302998176162308E-001,
```

```
    1.1020445173406628E+000, -1.9863209326779568E-001, -
5.2479399876888655E-001,
    3.1038109946620746E-001, 9.2763393603079125E-001 },
  -1.0,                                 /* Constant_Value :
'<S59>/Constant'
                                         */
  6.7276662990012670E-002,              /* Constant_Value_o :
'<S68>/Constant'
                                         */
  1.0,                                  /* Constant1_Value_n :
'<S59>/Constant1'
                                         */
  1.3968500572323897E+004,              /* Constant_Value_f :
'<S67>/Constant'
                                         */
  -1.0,                                 /* Constant_Value_p :
'<S60>/Constant'
                                         */
  4.4229867797250364E+002,              /* Constant_Value_n :
'<S70>/Constant'
                                         */
  1.0,                                  /* Constant1_Value_c :
'<S60>/Constant1'
                                         */
  3.4572752660981430E+004,              /* Constant_Value_m :
'<S69>/Constant'
                                         */
  -1.0,                                 /* Constant_Value_fr :
'<S61>/Constant'
                                         */
  1.9064422461636815E+002,              /* Constant_Value_l :
'<S72>/Constant'
                                         */
  1.0,                                  /* Constant1_Value_l :
'<S61>/Constant1'
                                         */
  3.1610698012074081E+004,              /* Constant_Value_k :
'<S71>/Constant'
                                         */
  -1.0,                                 /* Constant_Value_fv :
'<S62>/Constant'
                                         */
  1.3689509516578283E+001,              /* Constant_Value_j :
'<S74>/Constant'
                                         */
  1.0,                                  /* Constant1_Value_h :
'<S62>/Constant1'
                                         */
  8.1869249613346310E+003,              /* Constant_Value_h :
'<S73>/Constant'
                                         */
  -1.0,                                 /* Constant_Value_po :
'<S63>/Constant'
                                         */
  5.5987812793409645E+001,              /* Constant_Value_e :
'<S76>/Constant'
                                         */
  1.0,                                  /* Constant1_Value_a :
'<S63>/Constant1'
                                         */
```

```
  2.4134121891575874E+003,             /* Constant_Value_d :
'<S75>/Constant'
                                        */
  -1.0,                                /* Constant_Value_pu :
'<S64>/Constant'
                                        */
  6.0565986251732740E-001,             /* Constant_Value_g :
'<S78>/Constant'
                                        */
  1.0,                                 /* Constant1_Value_lp :
'<S64>/Constant1'
                                        */
  1.3964168602081420E+003,             /* Constant_Value_fu :
'<S77>/Constant'
                                        */
  -1.0,                                /* Constant_Value_c :
'<S65>/Constant'
                                        */
  5.5709963467951580E+000,             /* Constant_Value_ka :
'<S80>/Constant'
                                        */
  1.0,                                 /* Constant1_Value_j :
'<S65>/Constant1'
                                        */
  8.5839429557879055E+002,             /* Constant_Value_fc :
'<S79>/Constant'
                                        */
  -1.0,                                /* Constant_Value_lj :
'<S66>/Constant'
                                        */
  2.4231635126193563E-001,             /* Constant_Value_ex :
'<S82>/Constant'
                                        */
  1.0,                                 /* Constant1_Value_e :
'<S66>/Constant1'
                                        */
  6.0222665160411407E+002,             /* Constant_Value_i :
'<S81>/Constant'
                                        */

  /*  Subtractminx_Bias : '<S121>/Subtract min x'
   */
  { 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0 },

  /*  Dividebyrangex_Gain : '<S121>/Divide by range x'
   */
  { 5.1256546310300910E-001, 5.2868696225699585E-001,
5.0817438166710549E-001,
    5.0119257156400654E-001, 5.0648178889826145E-001,
5.0288104404833633E-001,
    5.0035535461486857E-001, 5.0376922201713226E-001 },
  2.0,                                 /* Multiplybyrangey_Gain :
'<S121>/Multiply by range y'
                                        */
  -1.0,                                /* Addminy_Bias : '<S121>/Add
min y'
                                        */

  /*  IW112_Value : '<S89>/IW{1,1}(2,:)''
   */
```

```
  { -8.6430992973089027E-001, -3.2235984510823495E-001, -
5.8904128109270715E-001,
    -6.3917608956452032E-001, 1.1123799418050431E-002,
4.0781723576446405E-001,
    -5.7275365497969133E-001, -9.1597776651485185E-001 },

  /*  IW113_Value : '<S89>/IW{1,1}(3,:)''
   */
  { 4.2028098422701061E-002, 1.1553935311691670E+000, -
3.5585570048259219E-001,
    9.3538496472334309E-001, 3.6305287561704547E-001,
2.6358570058280328E-001,
    5.2172500351815898E-001, 9.4649003062520687E-001 },

  /*  IW114_Value : '<S89>/IW{1,1}(4,:)''
   */
  { -5.5100884448645182E-001, 6.0051276194337111E-001,
7.1884692062308642E-001,
    1.1061815669583337E+000, 9.1903417951139377E-001,
1.9565717243545780E-001,
    -8.7451712849655594E-001, -6.2101295141131052E-001 },

  /*  IW115_Value : '<S89>/IW{1,1}(5,:)''
   */
  { -1.9217003939680108E-001, -1.0444292576415155E+000,
5.4411830207105094E-001,
    -6.4135776610967843E-001, -5.5957608290978056E-001,
2.9105932154345932E-001,
    2.1583567077118546E-001, 3.1020169910350914E-001 },

  /*  IW116_Value : '<S89>/IW{1,1}(6,:)''
   */
  { -8.6432119521217809E-001, -5.6095537918602612E-001,
5.6301355256179486E-001,
    -3.3121731228536166E-001, 7.0203995680532560E-001, -
9.1933052036845386E-001,
    -1.1596583910884882E-001, 1.7230586701389811E-001 },

  /*  IW117_Value : '<S89>/IW{1,1}(7,:)''
   */
  { -9.7416935105048497E-001, -1.2334316541914223E+000,
3.7254029661704302E-001,
    5.2251771886571285E-001, 5.0457804519523053E-001, -
3.1687024957606846E-001,
    7.6523964930348731E-001, 3.4965147046629275E-001 },

  /*  IW118_Value : '<S89>/IW{1,1}(8,:)''
   */
  { 3.5053362865024368E-001, -1.2798745768742953E-001,
9.3153225686299607E-002,
    -1.7799844882839796E-001, 1.1453909307356798E+000, -
7.1764294126769448E-002,
    2.6344630329645052E-001, -2.3196098655403755E-001 },

  /*  IW119_Value : '<S89>/IW{1,1}(9,:)''
   */
  { 4.8248963979944076E-001, -4.7922254087024113E-001,
1.0079201827527002E+000,
    -7.7489332678886547E-001, -9.0972115034386880E-001, -
5.3262987235596448E-001,
    1.3090923742798413E-001, 2.3998349179823331E-001 },
```

```
  /*  IW1110_Value : '<S89>/IW{1,1}(10,:)''
   */
  { -4.3862037958326983E-001, 6.5245581438564304E-001, -
6.1458638930585152E-001,
    -1.9679743564651181E-001, -2.5982843698674096E-001,
4.1565015999616683E-001,
    9.0581800675136381E-001, 6.2172840785663297E-001 },

  /*  IW1111_Value : '<S89>/IW{1,1}(11,:)''
   */
  { -5.7953454679552263E-001, 9.6587004662938092E-001,
3.2530519363825128E-001,
    1.5701368800647691E-001, 2.9887407792625409E-001,
7.2680445060876187E-001,
    -4.4543693410336443E-001, -9.6091168282777994E-001 },

  /*  IW1112_Value : '<S89>/IW{1,1}(12,:)''
   */
  { -2.5958814892875604E-001, 3.1396822443477074E-001,
1.5708933547846468E-001,
    9.4875963300595989E-001, -4.6343047235119378E-001, -
4.2254474316670704E-001,
    -2.5283344683777886E-001, -8.3167962109026283E-001 },

  /*  b1_Value : '<S84>/b{1}'
   */
  { 1.3527495323028857E+000, -1.0044713310239642E-001, -
1.3437816973819222E-001,
    2.1012848418228308E-001, -1.9040903853518817E-001, -
8.6626855513853651E-001,
    -8.8745385770789342E-001, -6.5994298004261431E-001, -
1.0986781722072119E+000,
    -7.0713087850087586E-001, -3.5353689345144473E-001,
1.1714523134324251E+000
  },
  -4.0314619793310297E-001,           /* b2_Value : '<S85>/b{2}'
                                       */
  -1.0,                               /* Gain_Gain : '<S117>/Gain'
                                       */
  1.0,                                /* one_Value : '<S117>/one'
                                       */
  1.0,                                /* Subtractminy_Bias :
'<S123>/Subtract min y'
                                       */
  0.5,                                /* Dividebyrangey_Gain :
'<S123>/Divide by range y'
                                       */
  2.0,                                /* Multiplybyrangex_Gain :
'<S123>/Multiply by range x'
                                       */
  -1.0,                               /* Addminx_Bias : '<S123>/Add
min x'
                                       */
  -1.0,                               /* Gain_Gain_n : '<S4>/Gain'
                                       */
  0.5,                                /* Constant1_Value_nh :
'<S3>/Constant1'
                                       */

  /*  IW211_Value_a : '<S45>/IW{2,1}(1,:)''
```

```
   */
  { 7.8922992552207338E+000, 4.9083339809886198E+001,
1.4939159839606132E+001,
    2.7229564279516467E+001, -6.5119513862360578E-001, -
3.8414256922073298E-001,
    -7.5585907460202173E-001, -1.2559976890238329E+000, -
6.5881751118096088E-001,
    5.3316681395780974E-001, -9.6317221689328285E+000,
1.9159380573046031E-001 },

  /*  IW111_Value_p : '<S18>/IW{1,1}(1,:)''
   */
  { 1.1289578937921581E+000, 3.1734706174258711E-004,
8.0242885008800247E-001,
    -3.6508008483885923E-001, -1.0692856248350746E+000, -
2.7007455254564272E-001,
    -1.0560707801400988E+000, -4.2946805200073629E-001 },

  /*  Subtractminx_Bias_f : '<S50>/Subtract min x'
   */
  { 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0 },

  /*  Dividebyrangex_Gain_p : '<S50>/Divide by range x'
   */
  { 5.0033573226356565E-001, 5.0279465125452638E-001,
5.0069467278071489E-001,
    5.0119257156400654E-001, 5.0059487021612770E-001,
5.0198333507987980E-001,
    5.0167766750617104E-001, 5.0319801689417532E-001 },
  2.0,                                 /* Multiplybyrangey_Gain_k :
'<S50>/Multiply by range y'
                                        */
  -1.0,                                /* Addminy_Bias_n : '<S50>/Add
min y'
                                        */

  /*  IW112_Value_n : '<S18>/IW{1,1}(2,:)''
   */
  { 1.6528754782248900E-001, -2.9937964570172021E-001, -
1.1779173882656216E-001,
    3.2008438760108399E-001, -6.3472008689764114E-001, -
4.8339507148896116E-001,
    2.8930917857448474E-001, 1.5176457427577306E-001 },

  /*  IW113_Value_k : '<S18>/IW{1,1}(3,:)''
   */
  { -3.0213540694012669E-001, 7.0318885339916437E-001, -
7.0611497720691963E-001,
    -2.4144717030807141E-002, -1.0734727333624972E+000, -
1.8785722940673155E-001,
    3.4113340615197413E-001, 7.5488416639475642E-001 },

  /*  IW114_Value_h : '<S18>/IW{1,1}(4,:)''
   */
  { -6.4347763193268481E-001, 1.3117863329088192E-001, -
4.3653585803098360E-002,
    5.6400481731709771E-002, -6.9771452877421869E-001, -
7.0567211624129067E-001,
    2.1948947086897239E-001, 7.5269008860755748E-001 },

  /*  IW115_Value_o : '<S18>/IW{1,1}(5,:)''
```

```
   */
  { -1.3932368256577077E-001, 8.4540425744172462E-002, -
4.6828593831518622E-001,
    -6.1157289002022097E-001, -6.3895801959436793E-001, -
6.2786825571987137E-001,
    1.7538779136125240E-001, 2.4869510480908194E-001 },

  /*  IW116_Value_b : '<S18>/IW{1,1}(6,:)''
   */
  { 1.6196907178913200E-002, 8.6004384866273953E-001, -
2.6588103311400119E-001,
    -1.2228751945868996E+000, -3.8777413098955932E-001,
1.0807145829867570E-001,
    -4.9301443808764728E-001, -2.6935089833469145E-001 },

  /*  IW117_Value_k : '<S18>/IW{1,1}(7,:)''
   */
  { 3.7075284759663857E-001, 2.4589601217209342E-001,
5.9807238616328184E-001,
    -4.7115586582855518E-001, -5.7414199856474524E-001, -
4.1687662193152775E-001,
    6.1550052957187706E-001, 7.7163354541404539E-001 },

  /*  IW118_Value_o : '<S18>/IW{1,1}(8,:)''
   */
  { -9.8909604600375622E-001, 2.8842313880595044E-001, -
2.6139741519577248E-001,
    -4.8475258354352076E-002, -6.4662818605112038E-001, -
7.1236848035171985E-001,
    -4.7317525045330683E-001, -4.9697926923463670E-002 },

  /*  IW119_Value_i : '<S18>/IW{1,1}(9,:)''
   */
  { 1.6817104194058469E-001, 5.4078006739012241E-001, -
2.8830286448177328E-001,
    -9.7414296806757106E-001, 2.7384509149941882E-001,
1.1447738514993713E+000,
    -1.8750245970739696E-001, 8.7421724343009444E-001 },

  /*  IW1110_Value_e : '<S18>/IW{1,1}(10,:)''
   */
  { 1.8897937363727013E-001, -4.4626919658866543E-001, -
2.1044747249912410E-001,
    1.0314298772974859E+000, 7.2604194060176208E-001, -
3.7422426361963579E-001,
    7.1108953970559796E-001, -9.9914747508995250E-001 },

  /*  IW1111_Value_g : '<S18>/IW{1,1}(11,:)''
   */
  { -6.2491619170945689E-001, -5.5086836793649208E-001, -
8.5281542343735584E-001,
    -9.4644489354042671E-001, 8.0573697792001375E-001,
6.7372768926502391E-001,
    6.3752052476726839E-001, -1.2572748752148906E-002 },

  /*  IW1112_Value_h : '<S18>/IW{1,1}(12,:)''
   */
  { 1.0321370728885266E+000, -2.2802022305213714E-002, -
4.1108487103213343E-001,
    2.4905656739286658E-001, -6.3370495888553335E-001,
2.8941166466751556E-001,
```

```
    -2.6590100271890449E-001, 3.6748840601455729E-001 },

  /*  b1_Value_i : '<S13>/b{1}'
   */
  { -8.7959371970838474E-001, -5.6513155574983853E-001, -
8.6277151020142884E-001,
    -5.9434304098611590E-001, 4.5910829812376992E-002,
3.3533000706589056E-001,
    -6.6432540708350318E-001, 3.4366167370984851E-001, -
2.1217978297303530E-001,
    -7.5519392388373896E-001, -5.1204072145691426E-001, -
9.2987850901181468E-001
  },
  -2.3819740735196357E-001,            /* b2_Value_a : '<S14>/b{2}'
                                        */
  -1.0,                                /* Gain_Gain_p : '<S46>/Gain'
                                        */
  1.0,                                 /* one_Value_j : '<S46>/one'
                                        */
  1.0,                                 /* Subtractminy_Bias_p :
'<S52>/Subtract min y'
                                        */
  0.5,                                 /* Dividebyrangey_Gain_c :
'<S52>/Divide by range y'
                                        */
  2.0,                                 /* Multiplybyrangex_Gain_o :
'<S52>/Multiply by range x'
                                        */
  -1.0,                                /* Addminx_Bias_l : '<S52>/Add
min x'
                                        */
  0.0,                                 /* Q_Y0 : '<S83>/Q'
                                        */
  1.0,                                 /* Q_Y0_k : '<S83>/!Q'
                                        */
  0U,                                  /* Switch1_Threshold :
'<S59>/Switch1'
                                        */
  0U,                                  /* Switch1_Threshold_d :
'<S60>/Switch1'
                                        */
  0U,                                  /* Switch1_Threshold_m :
'<S61>/Switch1'
                                        */
  0U,                                  /* Switch1_Threshold_h :
'<S62>/Switch1'
                                        */
  0U,                                  /* Switch1_Threshold_l :
'<S63>/Switch1'
                                        */
  0U,                                  /* Switch1_Threshold_mz :
'<S64>/Switch1'
                                        */
  0U,                                  /* Switch1_Threshold_e :
'<S65>/Switch1'
                                        */
  0U,                                  /* Switch1_Threshold_b :
'<S66>/Switch1'
                                        */
```

```
  0U,                                  /*
Nonzeroincrementcounter_Initial : '<S4>/Non zero =  increment
counter'
                                       */
  200U,                                /*
Nonzeroincrementcounter_HitValu : '<S4>/Non zero =  increment
counter'
                                       */
  0U,                                  /*
nonzeroincrementcounte_InitialC : '<S3>/non zero =  increment counte'
                                       */
  50U                                   /*
nonzeroincrementcounte_HitValue : '<S3>/non zero =  increment counte'
                                       */
};

/* Constant parameters (auto storage) */
const ConstParam_schemefinal schemefinal_ConstP = {
  /* Computed Parameter: IC
   * Referenced by blocks:
   * '<Root>/Buffer'
   * '<Root>/Downsample'
   */
  0.0,

  /* Computed Parameter: WindowSamples
   * '<S1>/Window'
   */
  { 0.0, 2.4076366639015356E-003, 9.6073597983847847E-003,
    2.1529832133895588E-002, 3.8060233744356631E-002,
5.9039367825822475E-002,
    8.4265193848727382E-002, 1.1349477331863150E-001,
1.4644660940672621E-001,
    1.8280335791817726E-001, 2.2221488349019886E-001,
2.6430163158700110E-001,
    3.0865828381745508E-001, 3.5485766137276886E-001,
4.0245483899193585E-001,
    4.5099142983521961E-001, 4.9999999999999994E-001,
5.4900857016478033E-001,
    5.9754516100806410E-001, 6.4514233862723103E-001,
6.9134171618254481E-001,
    7.3569836841299885E-001, 7.7778511650980098E-001,
8.1719664208182263E-001,
    8.5355339059327373E-001, 8.8650522668136844E-001,
9.1573480615127267E-001,
    9.4096063217417747E-001, 9.6193976625564337E-001,
9.7847016786610441E-001,
    9.9039264020161522E-001, 9.9759236333609835E-001, 1.0,
    9.9759236333609835E-001, 9.9039264020161522E-001,
9.7847016786610441E-001,
    9.6193976625564337E-001, 9.4096063217417747E-001,
9.1573480615127267E-001,
    8.8650522668136844E-001, 8.5355339059327373E-001,
8.1719664208182263E-001,
    7.7778511650980098E-001, 7.3569836841299885E-001,
6.9134171618254481E-001,
    6.4514233862723103E-001, 5.9754516100806410E-001,
5.4900857016478033E-001,
    4.9999999999999994E-001, 4.5099142983521961E-001,
4.0245483899193585E-001,
```

```
    3.5485766137276886E-001, 3.0865828381745508E-001,
2.6430163158700110E-001,
    2.2221488349019886E-001, 1.8280335791817726E-001,
1.4644660940672621E-001,
    1.1349477331863150E-001, 8.4265193848727382E-002,
5.9039367825822475E-002,
    3.8060233744356631E-002, 2.1529832133895588E-002,
9.6073597983847847E-003,
    2.4076366639015356E-003 },

  /* Computed Parameter: TwiddleTable
   * '<S10>/FFT'
   */
  { 1.0, 9.9518472667219693E-001, 9.8078528040323043E-001,
    9.5694033573220882E-001, 9.2387953251128674E-001,
8.8192126434835505E-001,
    8.3146961230254524E-001, 7.7301045336273699E-001,
7.0710678118654757E-001,
    6.3439328416364549E-001, 5.5557023301960229E-001,
4.7139673682599781E-001,
    3.8268343236508984E-001, 2.9028467725446233E-001,
1.9509032201612833E-001,
    9.8017140329560770E-002, 6.1232339957367660E-017, -
9.8017140329560645E-002,
    -1.9509032201612819E-001, -2.9028467725446216E-001, -
3.8268343236508973E-001,
    -4.7139673682599770E-001, -5.5557023301960196E-001, -
6.3439328416364538E-001,
    -7.0710678118654746E-001, -7.7301045336273699E-001, -
8.3146961230254535E-001,
    -8.8192126434835494E-001, -9.2387953251128674E-001, -
9.5694033573220882E-001,
    -9.8078528040323043E-001, -9.9518472667219682E-001, -1.0,
    -9.9518472667219693E-001, -9.8078528040323043E-001, -
9.5694033573220894E-001,
    -9.2387953251128685E-001, -8.8192126434835505E-001, -
8.3146961230254546E-001,
    -7.7301045336273710E-001, -7.0710678118654768E-001, -
6.3439328416364593E-001,
    -5.5557023301960218E-001, -4.7139673682599786E-001, -
3.8268343236509034E-001,
    -2.9028467725446244E-001, -1.9509032201612866E-001, -
9.8017140329560451E-002
  },

  /* Computed Parameter: MaximumCount
   * '<S4>/Non zero =  increment counter'
   */
  200U,

  /* Computed Parameter: MaximumCount
   * '<S3>/non zero =  increment counte'
   */
  50U,

  /* Computed Parameter: IC
   * Referenced by blocks:
   * '<S55>/Delay'
   * '<S58>/Delay'
   */
  1
```

```
};

/* File trailer for Real-Time Workshop generated code.
 *
 * [EOF]
 */
```

Related Publications

S. P. Le Blond and R. K. Aggarwal, *Impact of Wind Farms on Electro-Magnetic Transients on 132kV Network with Particular Reference to Fault Detection*, CIRED, Prague, 2009.……I

Le Blond, S.P.; Aggarwal, R.K.; Abdulhadi, I.F.; Burt, G.M.; , *Impact of DFIG wind farms and instrument transformers on transient based protection*, Developments in Power System Protection (DPSP 2010). Managing the Change, 10th IET International Conference on, vol., no., pp.1-5, April 2010…….....II

Le Blond, S.; Aggarwal, R.; *A review of artificial intelligence techniques as applied to adaptive autoreclosure, with particular reference to deployment with wind generation*, *Universities Power Engineering Conference (UPEC), 2009 Proceedings of the 44th International* , vol., no., pp.1-5, 1-4 Sept. 2009 ……III

S. P. Le Blond and R. K. Aggarwal, *Design of adaptive autoreclosure schemes for 132kV network with high penetration of wind: Part 1 – Real time modelling*, IEEE PES T&D, New Orleans, 2010……..IV

S. P. Le Blond, B. Ross and R. K. Aggarwal, *Design of adaptive autoreclosure schemes for 132kV network with high penetration of wind: Part 2 – Real time development and testing* (submitted to IEEE Transactions on Power Delivery March 2011, publishing pending)…….. …….....….…...V

IMPACT OF WIND FARMS ON ELECTROMAGNETIC TRANSIENTS IN 132KV NETWORK, WITH PARICULAR REFERENCE TO FAULT DETECTION

Simon LE BLOND
University of Bath – UK
S.P.Le.Blond@bath.ac.uk

Professor Raj AGGARWAL
University of Bath – UK
R.K.Aggarwal@bath.ac.uk

ABSTRACT

Adaptive autoreclosure has been extensively researched as a protection methodology for overhead lines, with well-known advantages over conventional autoreclosure. However, the effect of modern wind farms, specifically power electronics, on existing adaptive autoreclosure methods is unknown. Using the DIgSILENT software, a small part of the UK Generic Distribution System network is constructed as a test system and connected to built-in DFIG and full converter wind farm models. EMT simulations are carried out whilst varying the parameters known to affect single phase-ground fault voltage signatures. The Discrete Wavelet Transform is subsequently applied to these waveforms. Results show that adaptive autoreclosing schemes may need particular attention when designed for DFIG connected lines, although the traditional approach of signal processing and AI is validated since the effect of fault parameters have far more significance than the generating technology concerned.

I - INTRODUCTION

The use of wind generation is increasing globally. In the UK, government targets have set the proportion of renewables in the generation mix at 20% by 2020. Therefore, the impact of wind generation on all aspects of the power systems must be extensively investigated. It is unknown to what extent digital protection algorithms designed for conventional plant, including adaptive autoreclosure, will require modification for wind generation. However, it is well accepted that in future power systems, smart and fast numerical relays will play an important role in preserving security of supply. The need to meet growing demand, coupled with difficulties in building new network, will lead to decreased transient stability margins as operators push more power through the existing infrastructure. Adaptive autoreclosure (AA) is the relay feature whereby the faulted signal is used to diagnose the nature of the fault and designate the most appropriate reclose action. This usually consists of blocking reclosure for permanent faults and reclosing as soon as a transient arcing fault has extinguished. This would be a valuable facility in stressed future scenarios, minimizing transient fault clearance times, optimizing stability and preventing unnecessary shocks to generator shafts. Moreover, systems with larger permeation of wind power must remain connected through grid faults in order to prevent collapse i.e. have ride through capability. An adaptive autoreclose specially tailored to wind power would help fulfill this criterion by minimizing the number and duration of voltage dips due to reclose operations, and thus ensure smoother transient performance of wind turbines.

II – METHOD

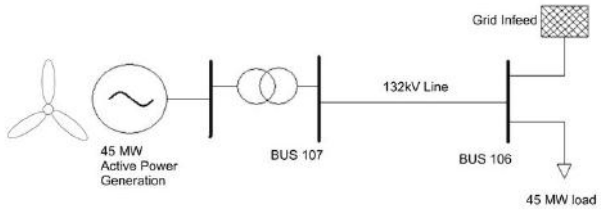
The study was conducted in DIgSILENT's powerfactory. Powerfactory is able to handle electromagnetic transient (EMT) studies and deals with load flow considerations demonstrating the viability of the power system over all timescales. The generating technology was a doubly fed induction generator (DFIG) based wind farm consisting of 9 x 5MW generators, and a 'full converter' (FC) wind farm consisting of 30 x 1.5 MW generators. In all cases, the wind farm was represented by one generator model connected in parallel, using the option in DIgSILENT 'number of parallel machines'. Simulations were compared to a base case of 30 x 1.5MW 'bare' synchronous generators. Since control circuitry governs output of reactive power in the case of the DFIG and FC turbine models, the generation capacities had to be defined in terms of nominal real power rather than apparent power.

i) Test network

The test network, shown in figure 2.1, was a small section of the UK generic distribution system (UKGDS) produced by the BERR funded SEDG centre [1], which is tailored for studies on distributed generation. The particular network used was the EHV2 "Large rural network". This was chosen since bus 7 is a convenient place to connect a hypothetical wind farm. The 13.3km 132kV branch from 106 to 107 represents one of the longest lines in the network representing the farm's likely remote location. Hitherto, adaptive autoreclosure methods have been extensively investigated for EHV transmission lines of 230kV and above. However, wind farms are usually connected at lower voltage levels at either sub transmission or distribution, and thus basing the investigation at 132kV was more appropriate. A load was placed at the receiving end bus. This was to satisfy DIgSIlENT's load flow component and ensure the line was transmitting power during the simulation. The rest of the grid was represented by DIgSILENT's external network component. The short circuit level of the grid infeed was set to a default value of 1000 MVA, approximately twenty times the capacity of the generation at the sending end. The line model was built up using DIgSILENTs 'Type Line' component, taking the values specified by the UKGDS. The line parameters for the

EMT study were calculated by DIgSILENT using a frequency-dependent distributed parameter model. The transformers connecting the generation were built into the DIgSILENT wind farm models, and modelled as ideal transformers, so as not to introduce extra transient behaviour. (This network is shown in figure 2.1).

Figure 2.1: Test network, bus 107 sending end 106 receiving



ii) EMT Simulation

Given the information used to determine reclosure would be from the fault itself and the action of the circuit breakers opening, the sequence simulated involved both the fault and subsequent opening of the circuit breaker. The transient simulation was run for one second, the fault inception was at 300ms from time zero and the circuit breakers were set to open at both buses at the nearest current zero after 340ms, with 2 cycles deemed a reasonable response time. The default fault type was a single phase to ground fault. This yields the most information post-circuit breaker by virtue of mutual capacitive and inductive coupling with the healthy phases. The fault path resistance had a default value of 2 Ω, and was simulated by connecting a phase to ground at a "virtual bus" at the middle of the transmission line. The default site of the fault, was at a distance of 6.65km, i.e. at a point equidistant to both ends. The voltage and current waveforms were measured from the sending end bus, where the protection relay would be located. The electromagnetic transient simulation in DIgSILENT was run for 1.1 seconds and at a sampling frequency of 10 kHz. This was a reasonable compromise between speed of simulation and accuracy. The waveforms were subsequently down sampled in Matlab to 1.6 kHz to define appropriate frequency boundaries for the wavelet transform. The resampling process in Matlab for each technology was uniform, so any error introduced by this was systematic.

iii) Discrete wavelet transform

Many adaptive autoreclosing schemes [2] consist of an initial signal processing stage, followed by a pattern selection algorithm. For signal processing, past schemes have utilized either the short time fourier transform or the discrete wavelet transform. The discrete wavelet transform offers the advantage of variable time and frequency resolutions. High frequencies favor time resolution, i.e. high frequency events specifically localized in time but poor frequency resolution. Conversely, low frequency signals have better frequency than time resolution. This property is useful for non-stationary power system transients that consist of localized high frequency information superimposed on the 50 or 60Hz fundamental power signal. A full explanation of wavelet transforms, and their application in power systems can be found in [3]. For this study the Daubechies DB4 wavelet was used to decompose the original sampling frequency of 1.6kHz to successive details. The frequency bands represented by the details were thus fixed at 800-400Hz, 400-200Hz, and 100-50Hz.

III – RESULTS

Adaptive autoreclose schemes must be robust enough to cope with the wide-ranging fault conditions that may occur on a line, without affecting the diagnosis in terms of phase selection, fault type or transient arcing time. Some of the most important factors that affect the fault signal are: capacity of sending end generation, receiving end short circuit capacity, length of transmission line, fault type, location of fault on line, point of fault inception on wave and fault resistance. For the purposes of this study, a further consideration can be added i.e., the effect of generating technology. [4]

i) Effect of generating technology

In the wavelet transform details, the main feature in all cases is an isolated, high intensity peak across all the frequency boundaries at the fault inception point, and then a smaller but equally ubiquitous spike at the point the circuit breaker opened. This is due to the near vertical wave-fronts on both fault inception and breaker opening, manifesting themselves in the transform as high intensity features at all frequencies. The differences between the FC and conventional technologies are unremarkable, with only minor variations in the frequencies after both events, in the 400-200 and 200-100Hz ranges. See figure 3.1 for conventional case and 3.2 for the FC, and their subsequent wavelet decompositions in figures 3.4 and 3.5. The results show profound differences between the DFIG (figures 3.3 and 3.6) and the other two technologies. With reference to 3.6, there is considerably more noise in the DFIG case over all frequency bands. This is due to the more complex control circuitry featured in the DFIG model, particularly the 'crowbaring' of the rotor current control circuit shortly after the fault inception. In the EMT simulation, the DFIG model automatically implemented this at approximately 0.307s, and removed the impedance at 0.807s. The distortion at the wavepeaks, following the fault inception, but before the A phase breakers open (see figure 3.3) is not present in the other two cases. This is most pronounced at the 200-100Hz and the 100-50Hz intervals, (figure 3.6). Over the healthy phases, the Full Converter model develops unstable oscillations 150ms after the fault (figure 3.7), resulting in overvoltages of up to 200% of the base. Comparison against real world data is required to determine whether this is down to

numerical instability in the DIgSILENT model or representative of actual behaviour.

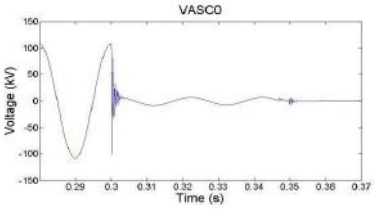


Figure 3.1: The conventional default case, faulted phase waveform

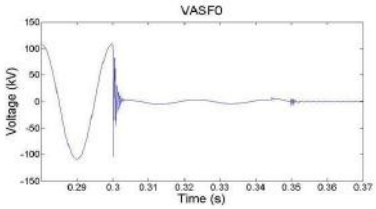


Figure 3.2: The full converter default case, faulted phase waveform

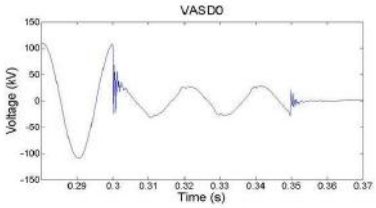
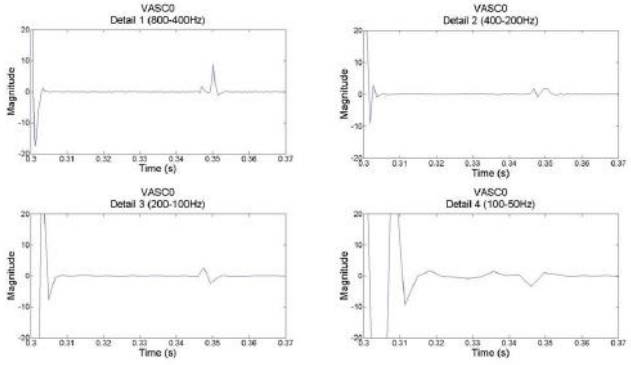
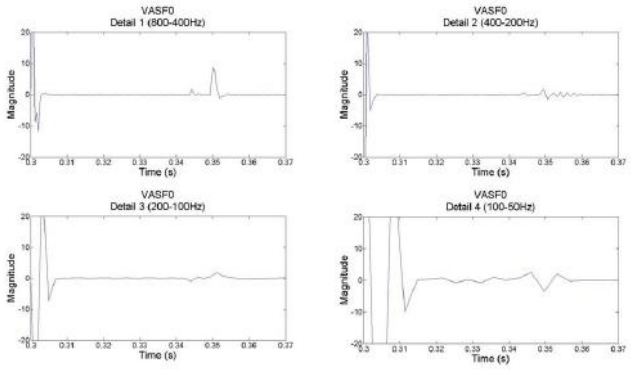


Figure 3.3: The DFIG default case, faulted phase waveform

Below, Figure 3.4: The wavelet transform details for the conventional default case, faulted phase waveform



Below, Figure 3.5: The wavelet transform details for the Full Converter default case, faulted phase waveform



Below, Figure 3.6: The wavelet transform details for the DFIG default case, faulted phase waveform

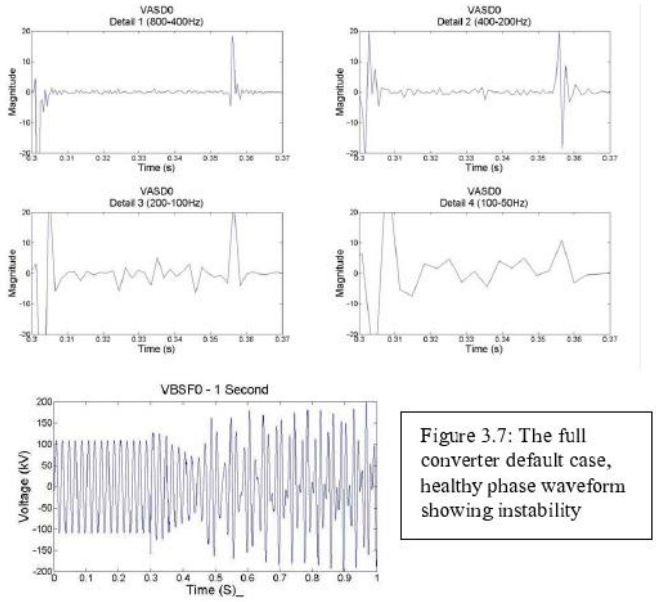


Figure 3.7: The full converter default case, healthy phase waveform showing instability

ii) Sending end capacity

The sending end generation was varied from 150MW to 10.5MW. The Full Converter shows very little variation in the transient signature. This is expected since the Full Converter is fully decoupled from the grid by power electronics and so can maintain current and voltage levels at a wide range of wind speeds and output power. The DFIG simulation fails for capacities higher than 45MW so it is only possible to compare this with 10.5MW. Despite this, the DFIG shows less variance in the fault signatures than the conventional case. This is likely due to the lack of current control circuitry in the conventional generation.

iii) Variance of short circuit capacity

The receiving end short circuit capacity was varied from 35GVA to 100MVA. As would be expected, the main effect across all generating technologies is the level of post fault voltage, which decreases as the local grid becomes weaker. The Full Converter and the conventional technologies demonstrate almost exactly the same behavior across all frequency boundaries. In the DFIG case, the 0.1GVA simulation is not stable. Since this configuration does not represent a valid mode of operation this result can be disregarded. With the strong grid, the features due to breaker opening are approximately ten times higher at the 400-200Hz and 200-100Hz bands in the DFIG case than in the other generator designs.

iv) Length of transmission line and fault location

The line length was varied from 50km to 5km. For all technologies, the transient fault response varies dramatically with the resistance. There are some subtle differences in the frequency bands between different generation types. The overriding trend is much quicker attenuation of high

frequency transients as line length is reduced. This is due to a smaller amount of trapped charge when the phase became isolated by the circuit breakers. This is common to all technologies so the implications for adaptive autoreclosure are minimal.

The location of the fault was varied from 100m from the receiving end to 100m from the sending end. Similarly, the less the amount of line between the measuring bus and fault, the shorter the oscillation time and thus less high frequency transients. There are some minor differences between the technologies, but these are far superseded by the effects of location of the fault. The greatest differences are in the DFIG due to the post-fault noise.

v) Fault type.

The fault type was altered to a three phase to ground, and the response of the circuit breakers adapted for a three phase trip. Although the trends are the same across all three phases, the B and C phases exhibit less intensity since their waveforms are interrupted at a phase shift of either plus or minus 120 degrees from the maxima. The DFIG exhibits very little high frequency information due to the initial fault or the subsequent operation of the circuit breaker compared to the other two technologies. It would be more useful to compare other more common fault types given this is the rarest type of fault to occur on a transmission line (less than 4%). Unfortunately, DIgSILENT does not allow fault analysis for one section of a distributed parameter line model so this would require alternative simulation software.

vi) Fault inception point on Wave

The timing of the fault was delayed slightly so it occured at half intensity and voltage zero. The effect across all technologies is a decrease in the magnitude of high frequency oscillations due a reduction in the step change. There are very small differences between the technologies apart from the characteristic high frequency noise from the DFIG, which is also observed in the default study.

vii) Fault Resistance

The fault resistance was varied from 0 Ω to 50 Ω. The high frequency features due to both the initial fault and the circuit breaker opening are suppressed more with increasing fault resistance. As can be expected, the post fault voltage level is also greater the higher the fault resistance. These effects are the same across all generating technologies. However, a constant resistance is only valid in the case of a permanent fault. It is well known that transient faults exhibit arcing behaviour, which have a non-linear time-varying resistance. Further studies to show how the electrical arcs interact with wind farms would be beneficial.

IV - CONCLUSIONS

The DFIG showed most variation. This is expected since it is not completely decoupled from the grid, and is therefore more sensitive to grid side transients [5]. In a number of cases, the DFIG simulation was unstable; this is likely to be due to numerical instability caused by the interaction of the complex control loops and the power system. In the DFIG design, the frequency converter must control current in the rotor to maintain synchronism with the grid, but shortly after fault inception the protective crowbar impedance disengages the rotor current control circuitry. The Full converter generally showed very similar variation to the conventional generation. This bodes well for adaptive autoreclosure since this concept is gaining an increasing amount of the market share. However, in many cases the healthy phase became unstable (figure 3.7), putting limitations on the duration of two phase operation on this design, and thus use of single pole circuit breakers. Further work would be beneficial on realistic arc models, and different fault types. Firm conclusions could only be drawn from comparisons with real world data or at least by replicating these results on other software. However, the traditional approach of using wavelet transforms with AI is validated. This is such because only the parameters that showed significant variance with wind generation are those known before the fault (i.e. grid capacity), and these can be accounted for 'pre-fault' in the relay settings. The random unknown parameters determined by fault type (e.g. location-related) had far more influence than the generating technology concerned.

Acknowledgments

This study is funded by Flexnet, part of the EPSRC led Supergen consortium: see www.supergen-networks.org.uk

REFERENCES

[1] BERR SEDGE (centre for Sustainable Electricity and Distributed Generation), *UK Generic distribution network*. 2004; [accessed 28/11/2008] Available from: http://www.sedg.ac.uk/ukgds.htm.

[2] D. S. Fitton, R. W. Dunn, R. K. Aggarwal, and A. T. Johns, 1996 "Design and implementation of an adaptive single pole autoreclosure technique for transmission lines using artificial neural networks", IEEE Trans. on Pow. Del., vol. 11, p 748-756

[3] Kim, C.H. and R. Aggarwal, 2001 "Wavelet transforms in power systems Part 2 Examples of application to actual power system transients" Power Engineering Journal,. **15**(4): p. 193-202.

[4] Johns, A.T. and R.K. Aggarwal, 1976 "Digital simulation of faulted EHV transmission lines with particular reference to very high speed protection" *Proc. of the I.E.E. -London,*. Vol 123(4): p. 353-359.

[5] T. Ackermann, 2005, *Wind power in power systems*. Chichester, England ; Hoboken (NJ): John Wiley, 2005, p68

IMPACT OF DFIG WINDFARMS AND INSTRUMENT TRANSFORMERS ON TRANSIENT BASED PROTECTION

S.P. Le Blond*, R.K. Aggarwal*, I.F. Abdulhadi†, G.M. Burt†

*University of Bath,UK, email s.p.le.blond@bath.ac.uk
† University of Strathclyde,UK, email iabdulhadi@eee.strath.ac.uk

Keywords: Transient response, wind power generation, instrument transformers, power system simulation, real time systems

Abstract

The impact of wind generation on high frequency, transient-based overhead line protection is investigated. An 8 bus section of the 132kV Scottish network with a realistic DFIG-based wind farm model is simulated on a real time digital simulator. Short circuit studies are conducted on a double circuit overhead line adjacent to the wind farm. Three case studies are compared: the DFIG model, a model with instrument transducers and the control model. Analysis is conducted on frequency and time domain voltage and current waveforms measured at the wind farm bus. These show that, compared with the control case, the impact of each case is slight, but the CVT attenuates the higher frequencies of the voltage waveforms slightly more than the DFIG case.

1 Introduction

Innovation in power system protection has an important role to play in the smart grid revolution. Smart grids must accommodate a more dynamic grid topology with bidirectional power flows, and an uncertain and changing generation mix. Transient based protection (TBP) could meet some of these challenges since it is immune to phenomena based at the power frequency; such as power swings and sub-synchronous resonance associated with compensation equipment. Most importantly, TBP could increase transient system stability and facilitate more power transfer, because of decreased critical clearing times due to extremely fast operating times. One of the few certainties in the future generation mix is an increased penetration of wind generation in order to meet legally binding emissions targets for 2020.

Unlike conventional generation, modern variable speed wind farms use power electronics for power conversion, the most common example being the Doubly Fed Induction Generator (DFIG). A possible concern in the deployment of TBP may be the switching frequencies degrading power quality, leading to deterioration in the performance of local TBP devices. It is therefore important to determine to what extent wind farms affect transient signatures used in TBP.

Novel protection techniques are commonly developed with a much simplified, two bus, power system model. These idealised models fail to capture the transient responses of components elsewhere in the system. Moreover, since high fidelity transient based simulations are computationally demanding they are difficult to execute in real time. Consequently these studies are unable to evaluate the longer term system response of automated control actions including those taken by protective relays. In this study, a section of the UK 132kV network (Fig 1.) is simulated in real time. Two DFIG-based wind farms are modelled along with the local network on a Real Time Digital Simulator (RTDS). The RTDS is a proprietary product from RTDS technologies. Parallel processing is used to accurately simulate a power system of arbitrary complexity in real time at frequencies up to 3kHz.

It would be useful to define at this point what is meant by *transient based protection* [1]. When the power system is subjected to a large step change, such as a short circuit fault, it will undergo a transient period of disturbance. On overhead lines, short circuits result in travelling waves emanating from the fault point and being reflected back and forth from the terminating bus bars. There is much information about the nature of the fault contained in these high frequency signatures. This information can be obtained by filtering or partial (windowed) frequency domain techniques such as the short time fourier transform and the wavelet transform. It can be used to trip circuit breakers by specially designed relays.

2 132kV Primary System

The system modelled forms part of the 132kV network in the Scottish highlands.

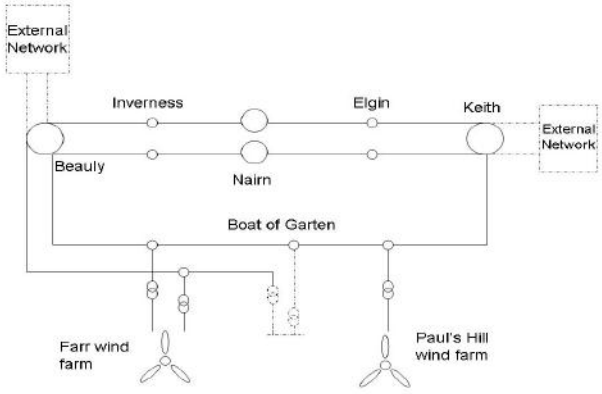


Fig 1: The modelled network can be considered the transmission system for this region.

The system comprises of eight busses arranged in a ring topology (see figure 1). 132kV is the highest voltage level so This area is ideal for this study because it contains two wind farms, that both use Bonus/Siemens 2.3MW DFIG machines. Farr wind farm consists of 40 machines with an installed capacity of 92 MW, and Paul's Hill, connected at the Glen Farclas bus bar, has 28 machines with an installed capacity of 64 MW.

An in-depth discussion of this model is available in [2]. Briefly, sources are represented using the RTDS source model behind the equivalent subtransient impedance. Although strictly speaking, the transient and steady state impedance should also be modelled, the timescales of concern in transient based protection are usually under 0.5 seconds so this approximation should not be too onerous. The over head lines (OHL) are represented using fully frequency dependent, distributed parameter models. Since all but one of the lines are of the double circuit type, it is important to represent the inter-circuit coupling as well as the inter-phase coupling. The RTDS line models were thus 6 conductor, three phase models with accurate placement of the conductors to represent the real world system. The loads were non dynamic and assumed to be purely inductive, modelled using equivalent shunt inductance and resistance. The load flow was specified according to National grid's peak load prediction for winter 2009 available in their seven year statement. The initial conditions are specified but settle down to a steady state after a few seconds of real time operation.

3 DFIG wind modelling

The specific control systems of wind turbines are proprietary to the manufacturer and so cannot be reproduced. The model included here is thus a generic model of a DFIG wind turbine developed by RTDS Technologies. The model's control systems are documented in [3]. The wind farm includes a mechanical model of the turbine, whose input wind speed, pitch and thus mechanical torque can be adjusted in real time. The switching of the valves for both the grid side and the rotor side VSC is decoupled and governed by two separate vector control schemes. This means that on the grid side, frequency can be maintained and real and reactive power can be independently controlled. On the rotor side, maximum energy capture over a wide range of wind speeds is achieved. The model of the partial power converter uses the small time step (below 2μs) VSC component of the RTDS. An interface transformer converts signals from the small time step module to the main power system time step (50μs). This is necessary for the fast switching resolution of the PWM voltage source converters. A discussion of how this is achieved in real time, and the interfacing with the main power system can be found in [4]. Ideally, the wind farm would be modeled using the full number of turbines and the cabled collector system, but this would require considerable processing power to achieve in real time. Therefore, a single turbine model has been scaled up to represent the installed capacity of the entire wind farm and connected at the relevant bus.

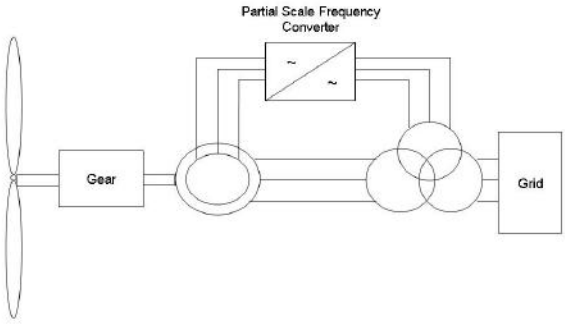


Fig. 2: Basic schematic of a DFIG

4 CVT and CT modelling

A concern with transient based protection is whether instrument transformers will faithfully reproduce the higher frequency system response. It is important therefore to model the CVT and CT with some accuracy, to assess their impact on the operation of transient based relays. The RTDS includes models for a capacitive voltage transformer (CVT) for measuring primary system voltages, and a CT model for system currents.

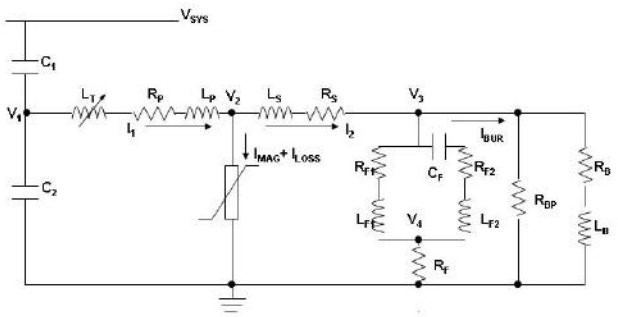


Fig. 3: Equivalent circuit for CVT model

The CVT model comprises of a string of capacitors in series with the primary system that may be adjusted to achieve the desired intermediate voltage V1. The equivalent circuit in figure 3 denotes capacitance C1 and C2. A tuning reactor, L_T, is placed at the intermediate bus to compensate the phase angle shift introduced by the capacitors. The intermediate voltage bus is then stepped down by a transformer to provide an acceptable input level, V3, for the relay, usually in the order of $115V_{L\text{-}L}$ RMS. In the RTDS model this transformer includes core hysteresis and saturation effects. A detailed explanation can be found in [5].

The RSCAD component library includes typical values for a 230kV installation. The capacitances C1 and C2 were changed to preserve the value of V1 in the presence of a different primary system voltage. The system voltage and the primary voltage are related by (1).

$$V_1 = V_{SYS} \frac{C_1}{(C1+C2)} \qquad (1)$$

In the RTDS model, the intermediate voltage V_1 was taken to be 17kV. According to [6] a typical value for the compensating inductor, L_c, is 42H for CVTs rated in the range of 110 – 500 kV. The value of L_c, the sum of C1 and C2, and the system frequency are related in the following way:

$$L_c = \frac{1}{\omega_0^{\,2}(C1 + C2)} \qquad (2)$$

Solving (1) and (2) for C1 and C2, gives respective values of 0.0538 μF and 0.1874 μF for the capacitors in the voltage divider.

The study CVT's desired output voltage V3, and the intermediate voltage V1, were identical to the conditions in the RTDS test model. It was also assumed that the transformer core saturation characteristics (supplied in the form of points on a B/H curve) would remain the same for a 132kV CVT. These values can be located in [5]

The simple CT model is based on the conductor passing through a toriodal core. The secondary side of the CT is formed by a number of turns around the core. The conductor forms the primary side and effectively is a single turn. The primary side resistance and inductance is negligible so can be ignored. The RTDS model has the following equivalent circuit shown in figure 4.

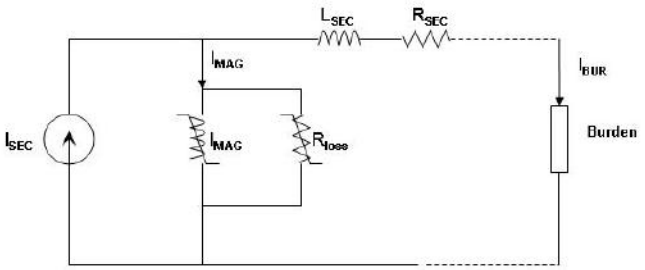


Fig. 4: Equivalent circuit for CT model

The magnetising branch only becomes significant when the CT core begins to reach saturation at the knee point. This may happen under certain transient conditions when large over-currents occur due to a fault. Under such conditions, an increasing amount of current is drawn down the magnetising branch and the current presented to the burden is (4).

$$I_{BUR} = I_{SEC} - I_{MAG} \qquad (4)$$

The RTDS model includes values for a typical installation with 200 turns on the secondary. Since the steady state current of the Beauly-Farr line varies with line loading from about 0.1 – 2kA, this would result in input level to the relay of 0.5-10A, which is reasonable. (Under fault conditions, the relay connected to the CT's secondary side would have to deal with higher overcurrents in the region of 20A. This upper input limit for the relay would depend on the short circuit capacity at either busbar, or, if the core became saturated under certain fault conditions, the knee point of the CT's operating region).

For both the CT and CVT RTDS examples all inductances and capacitances were scaled accordingly to give the same reactance for a 50Hz (rather than 60Hz) power frequency.

5 Methodology

When using transient signatures to affect a trip decision, the most important information is contained within the frequency spectrum immediately following a short circuit fault. The goal of the study was to establish how significant the presence of the wind farms was compared to the transducers. Thus several cases where investigated :

1 Control case: where the wind farms were modelled as equivalent sources and there were no transducers present

2 Instrument transformers: the voltage and current waveforms were obtained via the RTDS CVT and CT models

3 Wind farms: the wind farms at Farr and Paul's hill were modelled with the DFIG model outlined in section 3

From a TBP standpoint, there are further significant power system parameters that affect the transient fault response. These are the fault topology, the fault inception point on the waveform, fault resistance, location of the fault on the line, the power flow through the line and whether the fault is transient or permanent. For each case, seven fault topologies were considered, five with permanent fault resistances of 2ohms and two with variable arc resistance. All the other variables were kept constant with the default case, with the fault occurring at the mid point of the OHL and at voltage maxima on the waveform. Permanent faults were the single phase to ground, phase to phase, phase to phase to ground, three phase and three phase to ground.

Transient faults were modelled with the RTDS fault arc component. This models the variable resistance of an arcing fault according to the arc equation (5) discussed in [2].

$$\frac{dg}{dt} = \frac{1}{\tau}(G - g) \qquad (5)$$

Where g is the time dependant arc conductance, τ is the time constant and G is the stationary arc conductance. In the case of the primary arc, these variables are in turn based on constants that are determined empirically and a fixed arc length. The primary arc length for the single phase to ground fault was assumed to be the distance between the arcing horns 0.5m, and in the phase to phase case the distance between two phases 3.9m. The lower current 'secondary' arc model was not required since the signature under study was pre-circuit breaker. Readings were taken at the Farr bus on the Farr Beauly overhead line. This consists of a 30 km double circuit 132kV line with a single lynx conductor. The geometric tower layout can be found in the appendix of [2]. The CT and CVT waveforms were scaled to match the magnitude of the primary system waveforms.

5 Results

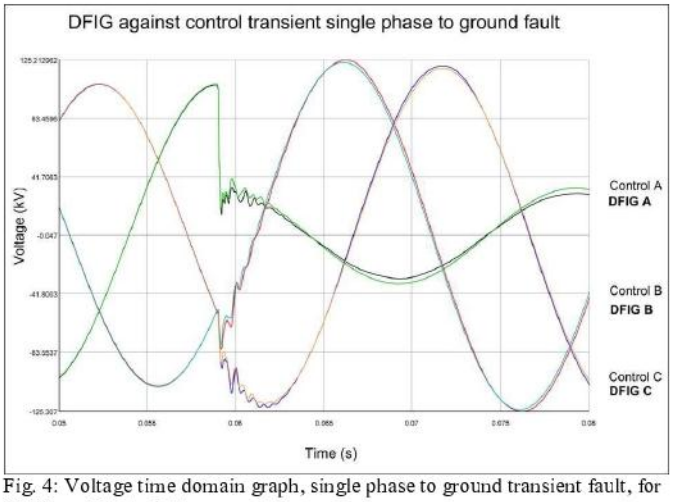


Fig. 4: Voltage time domain graph, single phase to ground transient fault, for DFIG and Control Case

In all cases the transients attenuated in under a quarter of a cycle. This was at voltage maxima fault inception point representing the greatest possible step change. For a transient single phase to ground fault in the time domain the DFIG case shows very little deviation from the control case, see figure 4. This is also the case for a permanent single phase to ground fault, see figure 5. Comparing these two figures indicates difference between arcing and permanent faults on voltage profiles is negligible.

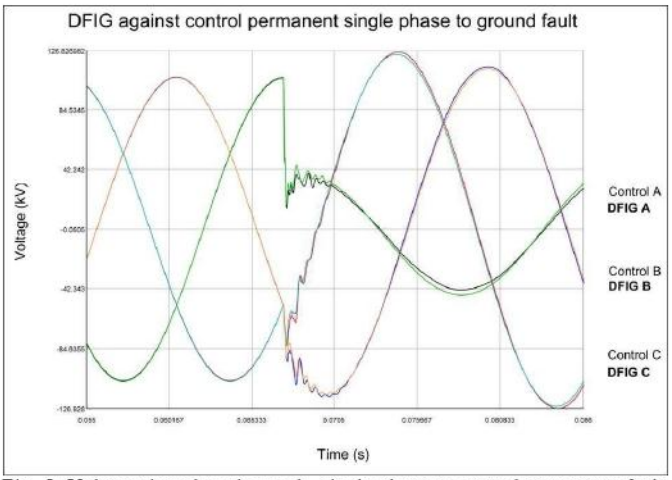


Fig. 5: Voltage time domain graph, single phase to ground permanent fault, for DFIG and Control Case

Figure 6 shows the fault current contribution of the DFIG case is approximately 0.66p.u. compared to the control case. Lower contribution to fault current is a recognised problem with DFIG machines, and has implications for power frequency protection. However, the transients here are very tenuous and cannot be directly compared in the time domain since post fault, the DFIG current is 180 degrees out of phase with the control case. Figure 7 shows that the CVT attenuates the higher frequency voltage perturbations in all phases.

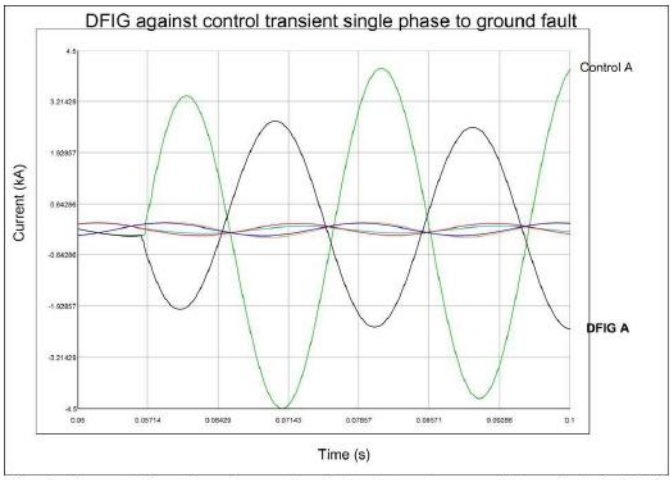


Fig. 6: Current time domain graph, single phase to ground transient fault, for DFIG and Control Case

This is expected as the CVT is inherently capacitive and thus acts as a low pass filter. The effect over the transient period is clearly greater than the DFIG and the control case, indicating that transient based protection robust enough to deal with CVTs could also deal with high penetration of DFIGs.

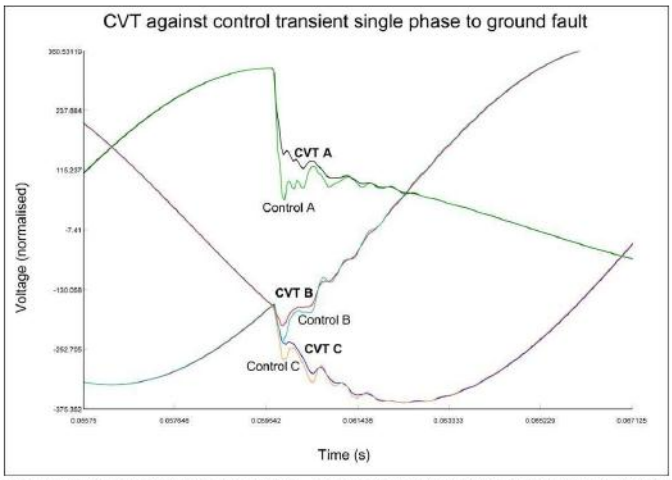


Fig. 7: Voltage time domain graph, single phase to ground transient fault, for CVT and Control Case

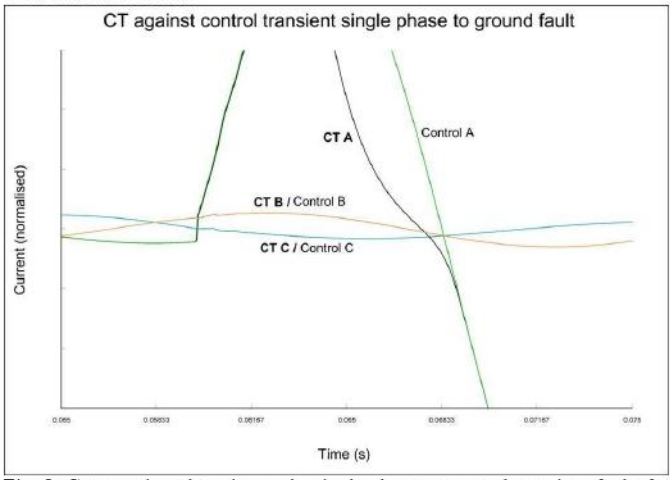


Fig. 8: Current time domain graph, single phase to ground transient fault, for CT and Control Case

Conversely, figure 7 shows the CT follows the high frequency current transients very well. It does become saturated for this high fault current, leading to a distorted faulted phase waveform that does not quite reach the peak with the primary current. (Figure 7 was scaled to show faithful transient CT performance so this lack of peak matching occurs off the scale). This has more significance in conventional power frequency relaying than TBP. The Fourier transforms in Fig 7-9 were compiled for each case for a single phase in a 3 phase to ground fault. Each transform was taken in a window half a cycle pre-fault and one cycle post-fault, and the magnitudes are plotted from 100 – 1000Hz on the same magnitude scale. In the case of the CVT, the time domain was scaled up to match the magnitude of the control waveform before the FFT was taken. Surprisingly, there is little variation in the frequency response of the three cases. The DFIG and the control case showed very similar frequency response, with the DFIG attenuating slightly more towards higher frequencies. As mentioned earlier the CVT acts like a low pass filter, but the cut off frequency of the filter depends on the CVT parameters. Here the CVT attenuates the higher frequencies only slightly more than the other two cases.

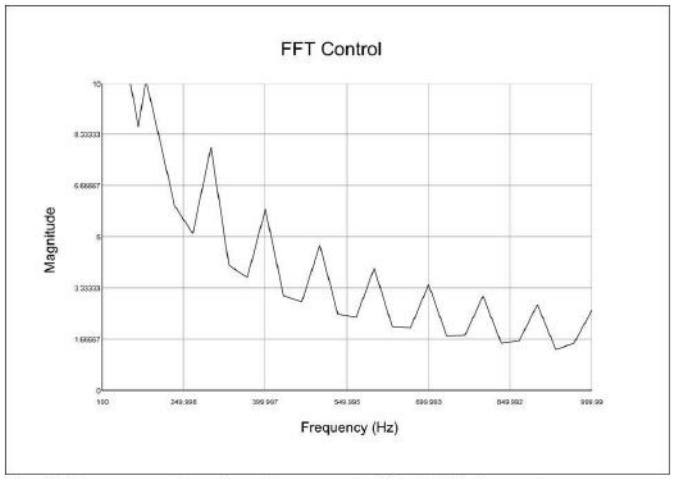


Fig. 7: Frequency domain voltage graph, 0.1-1kHZ, three phase to ground permanent fault, for Control Case

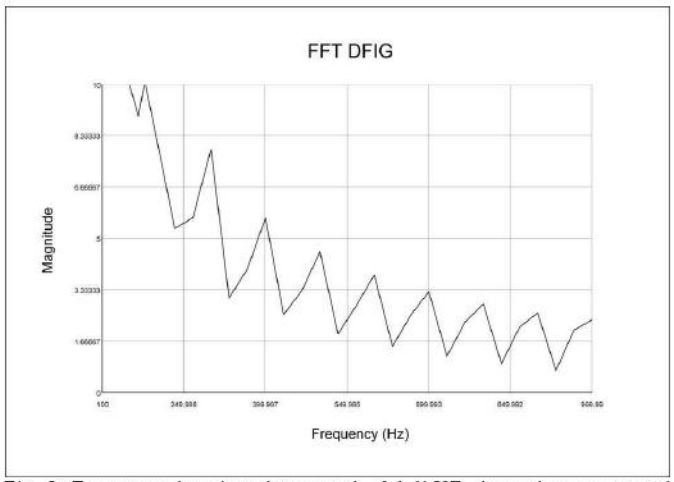


Fig. 8: Frequency domain voltage graph, 0.1-1kHZ, three phase to ground permanent fault, for DFIG Case

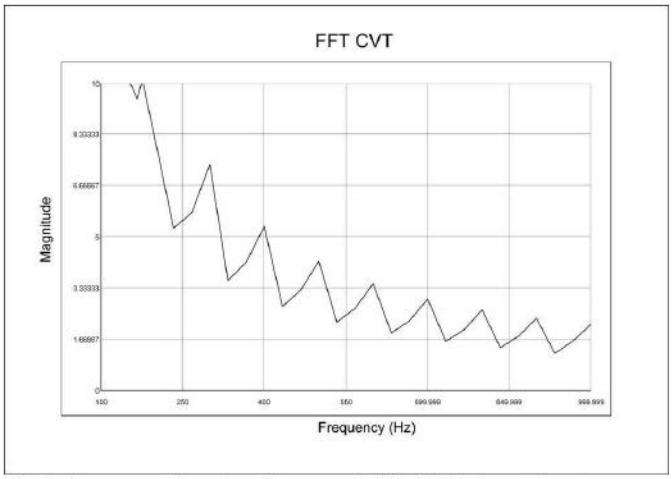


Fig. 9: Frequency domain voltage graph, 0.1-1kHZ, three phase to ground permanent fault, for CVT Case

6 Conclusions

An 8 bus system with wind generation has been simulated on an RTDS and used to investigate transient based non-unit overhead line protection. Very similar frequency responses in the cases bode well for transient based OHL protection. The CVT has more influence than the presence of DFIG wind farms. However, for this relatively short line length the transients attenuated in less than a quarter of a cycle. Sensitive signal processing would therefore be required in detecting this signature and discerning information in real time. This could however be achieved by fast DSP and AI techniques.

Acknowledgements

The authors would like to thank the EPSRC for funding provided as part of the Supergen: Flexnet consortium, grant number EP/E04011X/1.

References

[1] Z. Q. Bo, F. Jiang, Z. Chen, X. Z. Dong, G. Weller, M. A. Redfern "Transient Based Protection for power transmission systems," in *proc 2000 IEEE Power Engineering Society Winter Meeting - Vols 1-4*, pp. 1832-1837.

[2] S. P. Le Blond, R. K. Aggarwal, "Design of adaptive autoreclosure schemes for 132kV network with high penetration of wind: Part 1 - Real time modeling" (accepted for publication), IEEE PES T&D, New Orleans, USA 2010.

[3] R. Pena, J.C. Clare, G.M. Asher, "Doubly fed induction generator using back-to-back PWM converters and its application to variable-speed wind-energy generation," *IEEE Proc.1996 Electric Power Applications* vol. 143, pp. 231-241.

[4] T. Maguire and J. Giesbrecht, "Small Time-step (<2 microseconds) VSC Model for the Real Time Digital Simulator," presented at the Int. Conf. on Power Systems Transients, Montreal, Canada, 2005.

[5] RTDS Technologies, *Real Time Digital Simulator Power System Users Manual*, Winnipeg, Canada, 2006.

[6] Kojovic, L, Kezunovic, M., Skendzic, V., Fromen, C.W., and Sevcik, D.R., "A New Method for the CCVT Performance Analysis Using Field Measurements, Signal Processing and EMTP Modeling", *IEEE Transactions on Power Delivery*, Vol. 9, No. 4, pp. 1907-1915, October 1994.

A Review of Artificial Intelligence Techniques as Applied to Adaptive Autoreclosure, with Particular Reference to Deployment with Wind generation

Simon Le Blond
University of Bath
S.P.Le.Blond@bath.ac.uk

Raj Aggarwal
University of Bath
R.K.Aggarwal@bath.ac.uk

Abstract **- This paper presents a survey of artificial intelligence techniques that have hitherto been applied to adaptive autoreclosure, namely artificial neural networks, fuzzy logic and genetic algorithms. The aim is to discern the most suitable techniques for applying adaptive autoreclosure to systems with high penetrations of wind power. Traditionally, adaptive autoreclosure schemes have been implemented using a combination of signal processing and artificial neural networks. A number of variations on this conventional approach are proposed in this paper. Qualitative discussion shows that in theory, a combination of the examined AI techniques will provide the most robust methodology, combining the strengths of each technique whilst minimizing weaknesses.**

Index Terms **— Adaptive Autoreclosure, Power System Protection, Artificial Intellegence, Artificial Neural Networks, genetic algorithms**

List of abbreviations

AA = Adaptive Autoreclosure

SPG = Single Phase to Ground Fault

DAR = Delayed Autoreclosure

GA = Genetic Algorithms

IED = Intelligent Electronic Device

DFIG = doubly fed induction generator

WAMS = Wide Area Management System

I. Introduction

Adaptive autoreclosure is the relay facility whereby the faulted current and voltage waveforms are used to diagnose the most appropriate reclose action and time. It is well accepted that in future power systems, fast numerical relays will play an important role in preserving security of supply. The drive to meet increasingly ambitious renewable generation targets will lead to more distributed generation, and grids must evolve to accommodate this greater complexity with real time changes in topology due to the intermittent nature of wind. This imminent revolution in power systems is loosely referred to as 'smart grids'. In addition, the need to meet growing demand, coupled with delays in building new lines, will result in decreased transient stability margins as operators push more power through the existing infrastructure. Adaptive autoreclosure (AA) would be a valuable facility in stressed future scenarios, minimizing transient fault clearance times, optimizing stability and preventing unnecessary shocks to generator shafts.

As members of the Supergen flexnet research consortium, the authors are developing an adaptive autoreclosure scheme to cope with increased permeation of wind power in distribution networks. The scheme will form part of a numerical unit-based protection relay for overhead lines. Furthermore, autoreclose algorithms developed will be implemented within the coordinated Wide Area Management scheme [1] developed under flexnet at Strathclyde university.

Work completed so far [2] has demonstrated the transients introduced by power electronics in modern wind turbines, in particular the DFIG generator due to its complex control circuitry, will affect the frequencies used in adaptive autoreclosure schemes. This study highlights the need for robust diagnostic techniques for autoreclosing overhead lines connecting wind generating technology.

II. Adaptive Autoreclosure

Adaptive Autoreclosing (AA) schemes must discern the soonest time it is safe to reenergize a line without incurring secondary trips. Essentially this involves observing the extinguishing of the secondary arc in the case of a transient fault. A permanent fault, such as that caused by a downed line or a tree, will involve a constant fault resistance. A lightning strike or similar event leading to discharge to ground or between phases will be cleared by de-energising the faulted phase(s), (or all three phases should the circuit beaker not allow single pole reclosing). Such transient faults exhibit arcing phenomena, with the fault current flowing to ground through an electrical arc, usually between the arcing horns on a transmission tower. In this case, the fault resistance varies with time in accordance with the behavior of an electric arc. In the first instance, an AA scheme must make the distinction between a transient and a permanent fault through the presence of the arcing phenomena. If a permanent fault is diagnosed, reclosing must be blocked. In the case of a transient fault, the scheme must determine when it is safe to reclose. As outlined in [3] there is a short period after arcing has extinguished when reclosure will simply lead to further arcing. This is partly due to how the secondary arc re-strike

voltage interacts with the line voltage, and also in low wind conditions failure of the ionized arc products to clear can encourage re-striking over the original arcing path.

Over 70% of overhead line faults are single phase to ground (SPG) faults [4]. Circuit breakers that are capable of single pole opening will take advantage of this prevalence in fault types, and only de-energise one phase, thus preserving some of the power transfer through the line. Of course, this mode of operation can only occur for a few seconds since the unbalanced condition will lead to instability elsewhere in the system. Any AA scheme is heavily dependent on the type of circuit breakers employed since post fault information is required to assess the ongoing suitability of a reclosure decision. Therefore most adaptive autoreclosure schemes rely on secondary arcing that occurs after the faulted phase is opened, caused and sustained by coupling between the healthy and faulted phases. Three phase AA schemes have been proposed [5] for double circuit lines that use secondary arcs caused by inter *circuit* coupling.

In the UK, the vast majority of circuit breakers are three phase only. Thus AA schemes deployed on the current UK system must be designed accordingly. However, ageing infrastructure represents an opportunity to upgrade to single pole circuit breakers should this prove cost effective. Furthermore, much of the infrastructure connecting windfarms is yet to be built. The remote location of new substations will require a design emphasis on greater automation of secondary system and high and exposed overhead lines are more prone to transient faults caused by adverse weather conditions.

A further consideration is the effect of any reclosure decision on transient stability. A reclose operation causes a large step change in the configuration of the network, which follows the initial fault, also a large disturbance. It is thus important that any reclose operation be assessed from a stability point of view before it can be allowed. Moreover, the mismatching between mechanical power and electrical power can lead to extra stresses on generator components as the change in power flow is accommodated by rotor acceleration. Most modern turbines use power electronics to a greater or lesser extent in combination with induction generators so conventional treatments of transient synchronous stability are not valid. In this case stability is meant in the sense that wind turbines can tolerate grid faults without overspeeding. Results in [2] also suggest there is an upper limit to the length of time a full inverter turbine can remain in two phase operation.

With these considerations in mind, an adaptive autroreclosure facility must have the following abilities:

- If single pole tripping is employed, it must first be able diagnose the faulted phase
- It must distinguish between permanent and transient faults
- In the case of a transient fault, it must determine the safe autoreclosure time
- It must be able to reach a decision in real time
- It must not jeopardize stability or local generation through sudden changes in network topology
- The worst case reclosure decision must present no more threat to the integrity of the power system than a conventional DAR scheme.

III. Artificial Intellgence

Artificial intelligence is a branch of computer science that attempts to use examples in nature to solve problems that conventional computing techniques find problematic. Inspiration is drawn from a range of sources from human reasoning, natural selection in evolution and the physical topology of the brain. They share common qualities in that they are robust, highly-parallel, can deal with incomplete data, and perform well in extrapolating non-linearity. Escalation in complexity in power systems lends itself to problems to which artificial intelligence are well suited; examples include load forecasting, reactive power control, protection and alarm systems [6]. The complex interplay of variables that ultimately influence a reclose decision in adaptive autorelcosure is one specific area where AI techniques have been successfully applied.

IV. Artificial Neural Networks

A particular branch of AI is the Artificial Neural Network (ANN). An ANN comprises of a number of small processing components wired together, in a much simplified topology of the neurons and synaptic pathways in the brain. The processing components work in parallel increasing the speed of the device and tolerance to failure, in that a loss of a single processing unit will not significantly affect the functioning of the device. The synaptic pathways each have an associated weighting function and the neurons a firing function, known

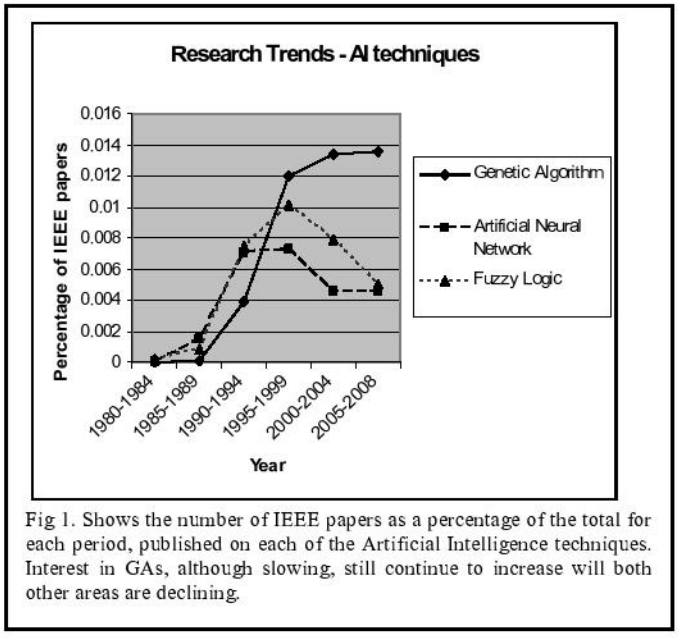


Fig 1. Shows the number of IEEE papers as a percentage of the total for each period, published on each of the Artificial Intelligence techniques. Interest in GAs, although slowing, still continue to increase will both other areas are declining.

as the transfer function. When the inputs to the neuron reach a level defined by the transfer function, the signal is transferred across to the next layer in the architecture. The weights must be adjusted to give the desired output for a given input. This process is known as training. Neural networks must be trained with data with a known outcome, in order to reach the desired outcome for unknown data. In practice this means that ANN based AA schemes must be subjected to a considerable number of fault cases before they can produce the correct output. The system under consideration must be simulated for a number of fault conditions offline before confidence in the ANN's performance can be assured. It is for this reason that such schemes in the past have been confined to bespoke solutions to a particular location on the network.

ANNs have been applied to AA schemes with some success. In particular, the Multi-Layer-Perceptron architecture has seen promising application in the past in [7] and [8]. However, using the raw time series data is impossible due to the many other power system characteristics that pollute the signal at a range of frequencies. The signatures benefit from processing in some way, so that through training, the ANN can weight the importance of contributions of different frequencies. Often preprossing is achieved by transforming the signal into the frequency domain, whilst retaining information where these crucial frequencies occur in the time domain. This can be achieved by the windowed FFT [8] and discrete wavelet transform [9].

Recent work on the transient response of windfarms [2] has validated the neuro-wavelet approach. This is because the fault conditions are shown to have a greater influence on the fault signatures than the generating technology. However, more work needs to be done on stability, and how the control system and design of the generating units in the wind influence this. If single pole operation proves unfeasible then perhaps the intercircuit coupling approach in [4] would be prudent.

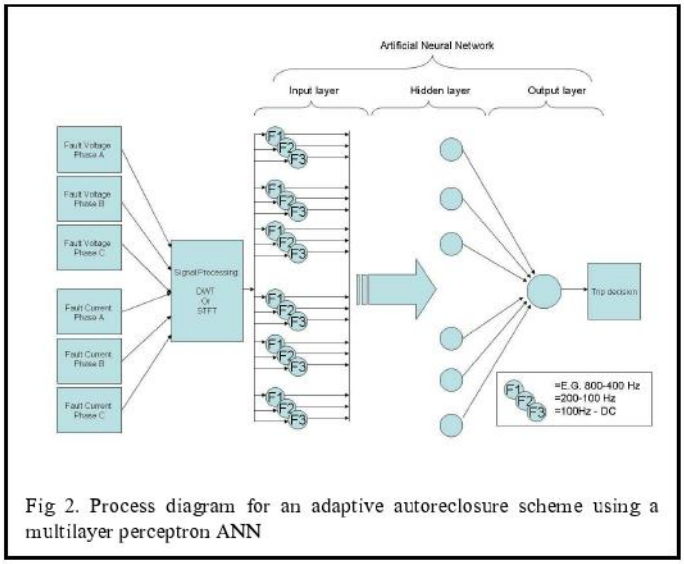


Fig 2. Process diagram for an adaptive autoreclosure scheme using a multilayer perceptron ANN

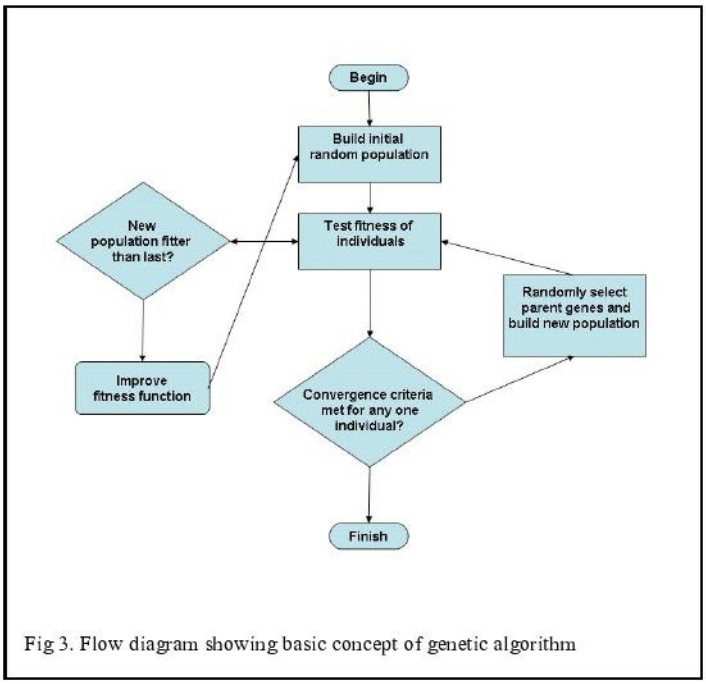


Fig 3. Flow diagram showing basic concept of genetic algorithm

V. Genetic Algorithms

Genetic Algorithms (GAs) are a subgroup of evolutionary computing along with genetic programming, classifier systems and simulated annealing. However, GAs are the most widely developed - they are behind over 95% of papers published in evolutionary computing [10]. Genetic Algorithms are a parallel search and optimization technique based on the mechanisms involved in evolution in nature. Their success is down to their simplicity and versatility concept. Figure 1 shows that their interest in the research community is continuing to increase. Genetic algorithms can be applied in a multitude of different ways, but are very useful for exploring a multidimensional search space whose dimensions are non-linear and non-differentiable. A genetic algorithm involves the same basic pattern:

1. Initially, a population of individuals, which each represent solutions to a given problem, is generated at random. The individuals, called chromosomes or genomes, are represented by "genes" that are elements to a solution. Each gene is usually encoded as a binary string.
2. These solutions are individually tested for their ability to solve the problem. They are then assigned a fitness based on this ability.
3. Two individuals are chosen at random to reproduce. This selection is weighted depending on the fitness score, so fitter individuals are more likely to be selected. The method used is thus often known as "roulette wheel" selection since it conceptually resembles a roulette wheel whose segments are different sizes.
4. The parents of the new individuals "reproduce" by each contributing some of their genes to the offspring. Crossover determines the amount of genes that each

individual supplies – usually the point in the genome where the parent genes are cut, and mutation represents the chance of a gene replicating error in the reproduction process. Offspring are then decoded and their fitness functions evaluated.

5. This process is repeated a number of times (successive generations) until a satisfactory solution is reached, i.e. an individual with a high enough fitness function is produced.

In addition to the processes described above, the fittest individuals can be cloned and included in subsequent generations in a process known as elitism. In some cases, the algorithm can improve the effect of the fitness function by assessing the new population's fitness against that of the previous generation. If the population does not improve, the fitness criteria can be made more rigorous. This mimics the predator-prey 'arms races' that occur in nature, where an advantageous change in the predator will stimulate a corresponding improvement in their prey.

The potential uses of genetic algorithms are wide ranging. However, their successful implementation depends heavily on defining and measuring the fitness function. Other critical factors are the length of the genome, population size, number of generations and the selection and encoding of the elements of the solution i.e. the genes. As such, algorithms in practice often require a good deal of 'tweaking' to maximize their efficacy.

In adaptive autoreclosure it is difficult to frame the reclose decision as a search problem because the full fault signature is not available. Also GAs require a large number of operations so are relatively computationally expensive, with long implementation times, even for modern processors. For these reasons, their use is generally confined to offline.

However, they have been used in the past to assist the training of MLP neural networks, and are shown to be more efficient than the standard back - propagation method [11].

Arguably the most important role of power system protection at transmission level is preserving stability. As mentioned before, the autoreclose scheme represents a large disturbance following an initial disturbance of the fault itself. Careful attention must therefore be paid to stability in any AA scheme. Transient stability margins in the UK are based on the interplay between tripping time following a fault, and the maximum power that can be transmitted without jeopardising synchronism should such a fault occur. Better clearance times allow greater power transfer. Calculations are often based on the assumption that stability is lost on the first power swing, but with fast reclose a second disturbance could also pose a threat to the system [4]. Safe reclose time depends on a whole array of variables depending on the local generation and network topology surrounding line on which it is deployed. This is further complicated with wind generation due to wind variation, asynchronous generators and power electronic converters. A genetic algorithm could be best used offline to generate a range of stable autoreclose sequences.

VI. Fuzzy Logic

Fuzzy logic, in the broadest sense, is a system of logic where elements occupy classes with boundaries that are not sharply defined [12]. It is often used in the form of expert systems, which rely on a series of IF-THEN-ELSE rules provided by expert knowledge to arrive at a decision. Fuzzy rule sets greatly enhance the performance of expert systems because degree of adherence to class allows built in confidence values rather than only allowing a variable to take on Boolean values. Similar to some ANN architectures, fuzzy logic rules can be used to map inputs onto nonlinear outputs. Unlike ANNs however, no training is required for calibration and the rules underpinning the networks are easy and intuitive to understand. Determining of the fuzzy rule sets can be achieved by the experience of a human expert rather than direct data relating to the problem at hand. In addition, rules can be added at a later stage without the need to rewrite existing rule sets. With an adaptive autoreclose facility this would mitigate the need for extensive simulation studies offline. It could also allow IEDs to be updated via remote link when improved fault data becomes available, or a user to extend the rule set manually should the system topology be altered with new plant.

The elements in the system are often defined with linguistic values rather than numeric variables. Although less precise, this emulates the uncertainty often inherent in real world problems and is more intuitive for humans. In fact, fuzzy logic is directly analogous to the way the human brain interprets language. Unlike conventional computing methods, where precision usually takes precedence over significance, fuzzy systems are able to manage this trade-off effectively. [6] In a fuzzy inherence system, sometimes only rough estimates of some quantities are required. Thus fuzzy logic lends itself to online calculations as computational expense is minimised.

In the past, fuzzy logic has often been applied in a hybrid system in conjunction with Neural networks [13] for the important task of phase diagnosis in single pole reclosure. This can be achieved in 10ms and so is effective as an online technique. Such a facility is a pre-requisite for any AA scheme based on single pole reclosing.

Moving towards commercial deployment, user installation of Adaptive Autoreclosing IEDs could potentially be improved by fuzzy logic. In such a scheme the user inputs data about the local primary system. A series of desired outcomes could be set based on the system – together with the 'worse case' limits as boundaries. For example, one such limiting rule could be "at all costs, an unsuccessful reclose operation should be avoided". Such inflexibility might not achieve optimal reclose times, but would nonetheless greatly improve them without the need for lengthy simulation and validation studies on the exact system configuration. A fuzzy scheme may also prove useful in selecting a pseudo-adaptive sequence, by applying human judgment in microseconds.

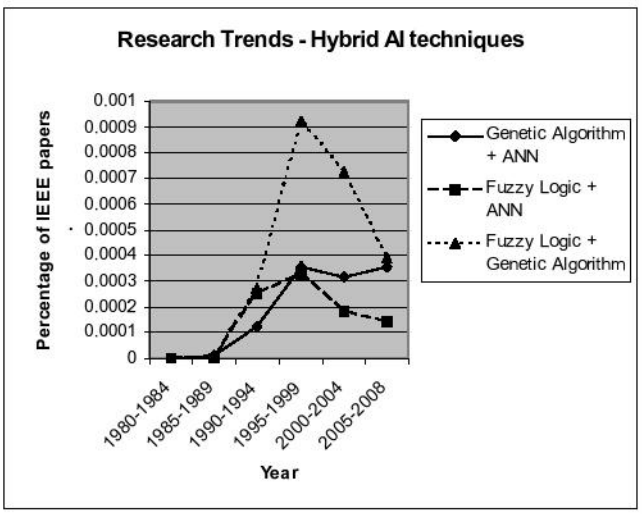


Fig 4. Shows the number of IEEE papers as a percentage of the total for each period, published on hybrid Artificial Intelligence techniques. Figures are found by cross referencing both of the terms shown.

VII. Hybrid AI Systems

In the past hybrid AA schemes have overcome some of the shortcomings presented by individual techniques. For example training of ANNs have been achieved with genetic algorithms [11]. Also recent work on the transient response of wind farms has validated the neuro-wavelet approach. Designing a practical AA scheme presents a series of challenges that are suited to different AI approaches.

For example, Fuzzy Logic rule sets can help in the installation of and calibration of IED by control engineers, who harness expertise on local network through providing a series of IF–THEN rules. (Also these may be easily updated at a later stage as knowledge is accumulated). As discussed previously, the complex interplay of variables that affect windfarm stability can be dealt with by genetic algorithms.

The exponential growth in microprocessor speeds and their decrease in costs in recent years gives feasibility to a scheme where a number of AI methods are employed in parallel. This would give a confidence value in any reclose decision issued, calculated on basis of level of agreement between different methods. Should the confidence value fall below a certain level, the reclose decision would default to a failsafe condition. Perhaps safest practice would be to run many algorithms in parallel and select the 'pre-programmed' sequence that the majority of the algorithms select; a kind of super-parallel "meta-neural-net".

It is hoped that in future the Distributed Protection Architecture (DPA), part of the WAMS system, under development at the university of Strathclyde [1] could incorporate adaptive autoreclose. It is also likely that AI techniques such as those discussed here will be implemented in the supervisory and management layers, AI scheme. With this approach an adaptive autoreclose action could be made with confidence in optimal system-wide consequences.

VIII. Conclusion

This paper represents a discussion of the considerations involved in adaptive autoreclosure schemes, and how AI methods have been used in the existing literature, as well as offering suggestions for future implementation.

Clearly each of the AI techniques discussed are suited to different applications. For this reason, a hybridisation of AI techniques is likely to build in greater robustness for future adaptive autoreclosure techniques.

Robustness is particularly important with lines near to wind generation. Due to the more complex control circuitry reclose decisions must take into account stability assessment along with restoration of supply.

Acknowledgement

The authors would like to thank all the contributors for flexnet consortium. Flexnet is an EPSRC led Supergen consortium: see www.supergen-networks.org.uk

References

[1] I. F. Abdulhadi, et al., "A Dynamic Modelling Environment for the Evaluation of Wide Area Protection Systems," UPEC - Universities Power Engineering Conference, Padova, Italy, 2008.

[2] S. P. Le Blond and R. A. Aggarwal, "Impact of Wind Farms on Electro-Magnetic Transients on 132kV Network, with Particular Reference to Fault Detection," CIRED, Prague, 2009, in press

[3] S. P. Websper, et al., "An investigation into breaker reclosure strategy for adaptive single pole autoreclosing," *IEE Proceedings-Generation Transmission and Distribution*, vol. 142, pp. 601-607, 1995.

[4] B. M. Weedy and B. J. Cory, *Electric power systems*, 4th ed. / B.M. Weedy, B.J. Cory. ed. Chichester: Wiley, 1998.

[5] R. K. Aggarwal, et al., "Adaptive 3-phase autoreclosure for double-circuit transmission systems using neural networks," 2nd International Conference on Advances in Power System Control, Operation & Management, Vols 1 and 2, pp. 389-392, 1993.

[6] K. Warwick, et al., *Artificial Intelligence Techniques in Power Systems*, IEE, London,1997.

[7] I. P. Gardiner and J. Ramsden, "On Site Experience of an Adaptive Autoreclose Relay for overhead lines," in IEE Conference on Developments in power system protection, March 1997, pp. 377-380.

[8] D. S. Fitton, et al., "The application of neural-network techniques to adaptive autoreclose in protection equipment," Fifth International Conference on Developments in Power System Protection, vol. 368, pp. 161-164, 1993.

[9] I. K. Yu and Y. H. Song, "Development of novel adaptive single-pole autoreclosure schemes for extra high voltage transmission systems using wavelet transform analysis," *Electric Power Systems Research*, vol. 47, pp. 11-19, 1998.

[10] M. Mitchell, *An Introduction to Genetic Algorithms*. 1st ed, Cambridge, Mass.; London: MIT, 1996.

[11] Y. H. Song, et al., "Genetic algorithm based neural networks applied to fault classification for EHV transmission lines with a UPFC," Developments in Power System Protection, Sixth International Conference on, 25-27 Mar 1997, pp. pp.278-281.

[12] Mathworks, (2004, Fuzzy Logic Toolbox Tutorial) Available: http://www.mathworks.com/access/helpdesk/help/toolbox/fuzzy/index.html, accessed 1/5/09

[13] H. Wang and W. W. L. Keerthipala, "Fuzzy-neuro approach to fault classification for transmission line protection," *IEEE Trans. on Power Delivery*, vol. 13, p. 11, 1998.

Design of adaptive autoreclosure schemes for 132kV network with high penetration of wind: Part 1 – Real time modelling

S. P. Le Blond, *Student Member, IEEE*, and R. K. Aggarwal, *Fellow, IEEE*

***Abstract*-- This paper is the first of a series of publications detailing the development an AI-based adaptive autoreclosing algorithm. In part 1, a detailed model of the Scottish 132kV Network has been constructed on a real time digital simulator. The system model is discussed, including the network topology, line and source modelling and the DFIG-based wind farm model. An initial investigation on penetration of harmonics from local wind farms is conducted using short circuit faults on two transmission lines in the network. This is necessary to ascertain to what extent wind farms may interfere with the adaptive autoreclosing scheme. As well as validating the model, the results suggest that penetration of harmonics is only significant on lines adjacent to the wind farms.**

***Index Terms*--Power system protection, real time systems, power system simulation, power system transients, power transmission lines, reclosing devices**

I. NOMENCLATURE

TBP = Transient based protection

OHL = Overhead Transmission Line

SPG = Single Phase to Ground Fault

VSC = Voltage Source Converter

PWM = Pulse Width Modulation

AI = Artificial Intelligence

AA = Adaptive Autoreclose

DFIG = Doubly Fed Induction Generator

Z' = **Sub-transient impedance**

$P_{s.c.c.}$ = **Short Circuit Power**

V_{l-l} = **Line to line voltage**

I_{peak} = **Bus three-phase peak current**

G = **Stationary arc conductance**

τ = **Arc time constant**

g = **Time dependant arc conductance**

S. P. Le Blond and R. K. Aggarwal are with the Centre for Sustainable Power Distribution, Department of Electronic Electrical Engineering at University of Bath, Bath, UK BA2 7AY (email s.p.le.blond@bath.ac.uk)

II. INTRODUCTION

Power system protection and control is an area to which the smart grid revolution brings opportunities and challenges in equal measure. In the UK in particular, an integral part of the short to mid term strategy for implementing a low carbon generation mix is a significant increase in wind generation. Latest government targets are 30% renewables in the generation mix by 2020 [1]. Recent advances in generation and power conversion have increased the scalability of wind power. Consequently wind farms will form some of the primary generators, along with many more embedded units connected at distribution level. From a technical perspective, the consequences of this for the UK system are two fold. The network operator must carefully consider the effect of high penetration of intermittent wind on maintaining the system, particularly stability, frequency response, power quality and voltage control. Larger farms may also have an impact on the correct operation of protection and control equipment. Increasing demand and the greater distance between generation and major load centres, will lead to a concomitant decrease in transient stability margins as greater power is pushed through the networks. From a protection standpoint, this demands a shorter critical clearing time, the time in which circuit breakers must operate in order to maintain system stability.

In recent years, the academic community has extensively investigated transient based protection (TBP) [2]. TBP uses the high frequency information, above the power frequency, within the transient fault signature to effect a relay decision. The advantages of such techniques are faster fault clearance times, immunity to electromechanical oscillations, power swings and sub-synchronous resonance associated with reactive power compensation equipment. Since more information is contained in the wideband signal, transient based relays are less vulnerable to mal-operartion caused by power system phenomena occurring at any localised frequency. In future, these techniques may be deployed in parallel with conventional power frequency based relays to increase selectivity, security and dependability.

Despite potential advantages, the uptake of TBP among manufacturers is slow. Power systems are safety critical so utilities are naturally reluctant to install novel relays over

proven techniques, and thus manufacturers have less incentive to develop and support them. Moreover, a significant technical barrier is the need to consider the unique transient response of each item of primary equipment.

Typically, novel unit-protection, including TBP, is developed using a much simplified power system model, with the component in between two bus bars, terminated in sources behind a subtransient impedance (see Fig. 1). However, when developing TBP, it is also important to model equipment outside the protected zone in order to be confident that it has no adverse effect on its performance.

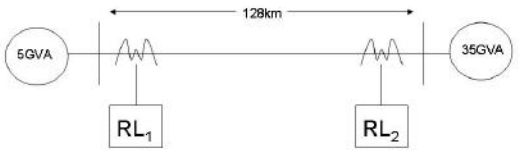


Fig. 1. A typical power system for investigating transient based OHL line relays.

Broadly speaking, the intended use of the test network described here is to develop a more realistic testing ground for TBP. The real world system will be used to evaluate the feasibility of TBP, and the effect of various power system components on TBP, including variable speed wind turbines.

This series of papers is concerned with the development of adaptive autoreclosing (AA) techniques. This is where the transient signature is used to diagnose the extinguishing of the secondary arc and thus the safe reclose times for temporary faults. Furthermore, reclosing is blocked for permanent faults minimising secondary shocks due to failed reclose attempts and leading to increased equipment lifetime. Since the signature associated with a dynamic arc resistance manifests itself at high frequencies this technique must be based on transients. The method developed will be building on those presented in [3]. This model will therefore initially be used to develop and demonstrate a transient based AA relay in real time.

III. Real Time Digital Simulation

The real time digital simulator (RTDS) is a proprietary product of RTDS Technologies. It uses parallel processing techniques on rack-mounted processors to maintain continuous real time digital simulation of a power system of arbitrary complexity. The computation techniques are based on those developed by H.W. Dommel and used in the well-known EMTP software [3]. The advantage of real time operation means that the power system operates in its own closed loop. The user may interact with the simulation in real time observing the effect of control actions.

The power system is drafted offline in a CAD-based program, 'RSCAD', and then uploaded to the RTDS hardware in real time via RSCAD's runtime module. Here fault condition(s) can be applied and the long-term power system response can be analysed and observed. If necessary, the user can interact with the simulation in real time via various control actions. These features combine to make a highly realistic simulation of a power system. The simulation time step is typically between 50-60μs, meaning that for real time operation, the RTDS must be capable of solving system conditions in under that time *for every successive time step.* RTDS Technologies state frequencies of up to 3kHz can be reproduced with confidence [4].

III. Network Model - 132kV Primary System

The modelled network forms part of the 132kV network in the Scottish Highlands. Currently, this network is owned and maintained by SHETL, part of the Scottish and Southern group, and operated by National Grid, the UK system operator.

The network comprises of eight 132kV bus bars in a ring topology as shown in Fig. 2. The northern branch of the network runs 90km from Beauly to Keith, supplying the load centres of Inverness, Nairn and Elgin. To the south, a 50 km branch supplies a load at Boat of Garten collecting power on route from the Farr wind farm. Another 50km branch runs from Boat of Garten to Keith via Paul's Hill wind farm, covering an area of approximately 1300km^2. Although in the UK, 132kV is considered subtransmission, here the network forms the transmission system since there is no higher local voltage network installed.

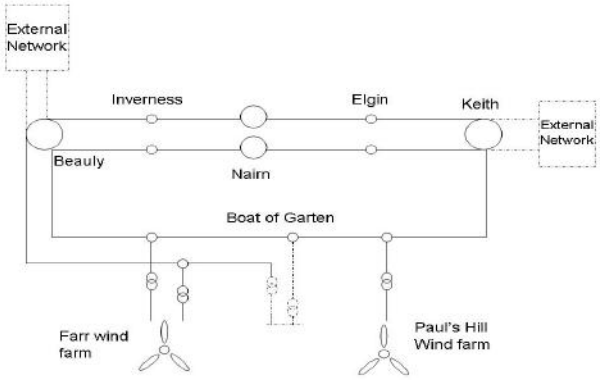


Fig. 2. Section of Scottish system under study, dashed components are omitted from the simulation.

The area is ideally suited for the study since it includes two wind farms, provides an area large enough for extensive real world studies, yet small enough to implement a real time transient model with the processing power available. In addition, an important constraint on travelling wave models is that the travel time must be longer than the simulation time step. In this network all the lines are longer than 15km, (a distance limitation imposed by RTDS's 50μs time step) so they may be represented using travelling wave models. Since the entire network is at 132kV, extensive transformer

modelling is not necessary, freeing up valuable processing power.

A) Source Modelling

The grid infeed points at Beauly and Keith are represented using the RTDS source model behind an equivalent subtransient impedance. These are based on the short circuit values for the three-phase peak current at each bus bar, taken from National Grid's seven year statement [5]. The subtransient impedance is calculated using (1) and (2)

$$P_{s.c.c.} = (\sqrt{3}V_{l-l})I_{peak} \quad (1)$$

$$Z' = \frac{P_{s.c.c.}}{I_{peak}^{2}} \quad (2)$$

The X/R ratio is then taken to be 13.5, an average of the values for 132kV quoted in [6]. This then gives the positive sequence impedance and phase angle that may be entered directly into RSCAD's source component. Strictly speaking, the sources should be represented by a dynamic impedance, where the subtransient reactance becomes the transient value and steady state reactance about 0.5 and 2 seconds post-disturbance respectively. This is beyond the scope of the paper due to limited information and computational power. However, since the phenomenon under consideration is mostly in the fast transient range, under 0.5 seconds, this approximation is reasonable.

B) Line Modelling

132kV overhead lines in this part of the network are predominantly of the double circuit tower type, typical of the UK system. The exception is the line connecting Keith to Boat of Garten, which is a single circuit. The conductor is the *Lynx* type with a single conductor per phase on all lines. The lines are not long enough to require transposition so the RTDS model was set to reflect this. All the line configurations can be found in appendix A.

The RSCAD-line modelling program allows RLC-type data entry. The information associated with each circuit is public domain and can be found in [5]. However, for the greatest accuracy and transient frequency response RSCAD requires physical data associated with the lines, including conductor configuration. The line data for the network can be found in appendix B, and is reproduced with permission of Scottish and Southern Energy.

The RTDS line models are distributed-parameter and frequency dependent, based on travelling wave theory. The line parameters are represented using hyperbolic functions in the frequency domain and then transferred to the time domain using convolution and the inverse Fourier transform. For simplicity, the assumption is made that the transformation matrix is frequency independent when this is not the case in reality. This does introduce some degree of error at DC and low frequencies, as the transformation frequency is generally chosen to be higher than the power frequency. However, the study is not concerned with sub-synchronous frequencies at this stage so this limitation should not be too onerous. For the double circuit lines a six conductor model is used which is able to determine inter-circuit coupling as well as inter phase coupling. A detailed discussion of travelling wave theory and its application for RTDS models can be found in [4].

C) Loads and Power flow

Loads are assumed to be purely inductive, and are modelled using a passive shunt inductance and a resistance connected at the relevant bus bar. The real and reactive loads were aggregated, based on the data in National Grid's estimated peak power flow for winter 2009 [5]. The true dynamic nature of loads was neglected since they are not critical, either at this voltage level or over the short timescales under investigation.

In RSCAD, initial conditions must be specified for the network sources and the generators within the network. These then settle down to a steady state after the simulation begins real time operation. Initial conditions were based on the peak power flow for winter 2009 published in National Grid's seven year statement [5].

D) Wind Farm Model

The Farr wind farm is located approximately ten miles south of Inverness. It consists of 40 Bonus/Siemens 2.3MW wind turbines, to give a total installed capacity of 92MW. Paul's Hill wind farm lies 23km southwest of Elgin and with 28 x 2.3MW turbines of the same manufacturer and model. The generators in the turbines are of the modern variable-speed DFIG type. The DFIG (doubly fed induction generator) is an induction generator where partial power conversion is handled by back-to-back converter.

The induction generator's rotor is connected to a partial frequency converter via slip rings, which in turn is coupled to the grid through a three-winding transformer. The stator of the generator is connected directly to the grid as shown in Fig 3. The power electronic converter makes up for the shortfall or excess speed difference (and thus the difference in the turbine's mechanical frequency and the grid's electrical frequency) by injecting the appropriate variable current into the rotor. In the over-synchronous case, power flows from the rotor to the converter to the grid, and in the sub-synchronous case it flows in the opposite direction. However, in either case, net power flow is onto the grid via the stator. This mechanism enables the turbine to operate at a wide range of speeds, typically up to +30% and -40% of synchronous speed.

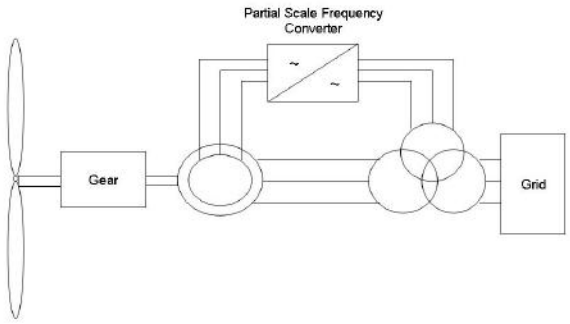


Fig. 3. Basic diagram of DFIG turbine.

The back-to-back converter allows two-way power flow, consisting of a rectifier and an inverter whose thyristors are Pulse Width Modulated. Fast discrete switching of the thyristors, controlled by modulating the width of signal pulse approximates a DC input signal to a sinusoid and vice versa. The basic premise is that power is taken at one AC frequency, converted to DC and then converted back to AC at the required grid frequency. The capacitor across the DC link allows control of the rectifier and inverter to be decoupled. This means more complex control of the thyristors can be introduced to suit the requirements both at the grid side (i.e. reduce power fluctuations and control voltage) and at the generator side (appropriate excitation currents can be established along with the desired rotor speed). It is the fast switching of the semiconductor thyristors that introduces harmonics onto the grid [7].

Unfortunately, transient models of the particular turbine design are not available since they are proprietary to the manufacturer. However, a generic model of a DFIG wind turbine has been developed by RTDS Technologies. A more in-depth explanation of the model can be found in [8]. Briefly, the wind farm includes a mechanical model of the turbine, whose input wind speed, pitch and thus mechanical torque can be adjusted in real time. The switching of the thyristors for both the grid side and the rotor side is decoupled and governed by two separate vector control schemes. This means that on the grid side, frequency can be maintained and real and reactive power can be independently controlled. On the rotor side, maximum energy capture over a wide range of wind speeds is achieved. The model of the partial power converter uses the small time step (below 2μs) VSC component of the RTDS. An interface transformer converts signals from the small time step module to the main power system time step. This is necessary for the fast switching resolution of the PWM voltage source converters. A discussion of how this is achieved in real time, and the interfacing with the main power system can be found in [9] and [4]. Ideally, the wind farm would be modelled using the full number of turbines but this would require considerable processing power. Therefore, a single turbine model has been scaled up to represent the installed capacity of the entire wind farm. It should be noted that for the harmonic study in section 4, the windfarms were represented by ideal harmonic sources, with the intention to use this model in later publications.

IV. Network Model – Secondary System and Miscellaneous

A) Transducer Modelling

An important aspect of designing transient based protection schemes is including the transient response of the instrument transducers. The RTDS includes models for a capacitive voltage transformer (CVT) for measuring primary system voltages, and a Current Transformer (CT) for system currents. Modelling the transient response of transformers is necessary since some conditions may initiate ferroresonance and inrush phenomena that may confuse or compromise relay functionality. A detailed discussion and an investigation into the effect of these models will be reported in part two of this paper.

B) Arc Modelling

In adaptive autoreclosing schemes it is necessary to determine whether a fault is transient or permanent. Transient faults exhibit arcing behaviour, with high frequency signatures due to a dynamic resistance. In the design of the scheme it is necessary to simulate the arc and its interaction with the power system, so that the logic may respond appropriately to an arcing fault or a permanent fault.

The arcing behaviour can be described by the primary and secondary stages. The primary arc is in the period before the circuit breakers open and is due to the fault current flowing from the energised phase to ground. The lower current secondary arc is sustained by the mutual coupling between the faulted and healthy phases, and only present when one or more of the phases remain energised. The behaviour of both arcs are governed by time varying conductance, and can be described by (3) the dynamic equation for unconstrained arcs in air.

$$\frac{dg}{dt} = \frac{1}{\tau}(G - g) \qquad (3)$$

Where G is the Stationary arc conductance, g is the time dependent arc conductance and τ is the time constant. The secondary arc is more complex because it has a varying arc length, on which the time constant depends, and a number of successive partial re-strikes. Arcing signatures also may be significant as a result of the action of the circuit breakers. Since the RTDS circuit breaker model operates at a current zero crossing it is assumed that no arcing takes place. Arc models and their implementation into the network model will be discussed in detail in part two of this paper [10].

V. HARMONIC PENETRATION STUDY

A suitable starting point for the study is to determine the extent to which harmonics generated by the wind farms penetrate onto the local grid. This is important for adaptive autoreclosing since harmonics may mimic high frequency information associated with the arcing fault, leading to a permanent fault being misdiagnosed as transient. A wind farm must comply to operating constraints imposed by the utility, known collectively as the *grid code*. An important aspect of this is ensuring a certain degree of power quality. The UK grid code demands all generators to be compliant to Engineering Recommendation G5/4 [11], which in turn is based on the international standard IEC61000. This governs the levels of harmonic distortion permissible, up to and including the 50th harmonic. ER G5/4 recommends the following planning levels for Harmonic voltages in systems between 20kV and 145kV [11].

TABLE 1
G 5/4 RECOMMENDATIONS FOR THE FIRST 9 HARMONICS ON SYSTEMS BETWEEN 20KV AND 145KV

Harmonic Number (h)	Harmonic Voltage %	Harmonic Number (h)	Harmonic Voltage %
2	1	6	0.5
3	2	7	2
4	0.8	8	0.4
5	2	9	1

In practice, wind farms have filters to suppress the level of harmonic distortion and the operational levels should be well below the planned levels. However, as a worst case scenario, harmonic emissions should not exceed those stated in G5/4. The harmonic source component in RSCAD allows up to four harmonics superimposed on the fundamental AC sinusoidal source. The magnitudes of the harmonics are expressed as a percentage of the fundamental. In order to assess the worse case scenario, harmonic emissions equivalent to G5/4 levels were used to represent both wind farms. (The DFIG wind farm model discussed earlier will be used for subsequent investigations). It has been observed with PWM back to back converters, the 5th and 7th harmonics are most significant occurring at 250Hz and 350Hz respectively [12]. The four harmonics were chosen to be the 3rd , 5th, 7th and 9th with respective harmonic voltage levels of 2%, 2%, 2% and 1%, as per G5/4.

A) Simulation Setup

As mentioned before, of particular interest to the authors are transient based adaptive autoreclosing schemes. During a single phase to ground fault, the current and voltage waveforms were measured at the OHL terminating bus bars in front of the breakers, since this is where the transducers will be based in practice.

Short circuit studies were conducted on the Beauly/Farr line, defined as "adjacent" and the Nairn/Elgin line defined as "distant". These lines were chosen since the former was adjacent to the wind generation, whereas the latter is the most distant. Also the lines are of a similar length, both suitable for phase domain travelling wave models on the RTDS. For each line, an equivalent control case was conducted for comparison. In the control cases the wind farms were represented by sources without harmonics.

For each case, a single phase to ground fault was initiated at the mid point of the line and the corresponding phase circuit breakers subsequently tripped at either end of the line simultaneously. The fault path resistance to ground was assumed to be 2 ohms. Adaptive autoreclosure schemes make use of frequency domain information. There is greater frequency domain information when single pole tripping is used over three phase tripping due to mutual inductive and capacitive coupling between the faulted and healthy phases. Although three-phase tripping is used extensively on the UK system, the circuit breakers were set to trip a single phase for the purposes of this investigation. (However, Adaptive autoreclosing for three phase circuitbreakers may be possible on UK type double circuit lines by using the signature arising from inter-circuit coupling. This technique will be investigated in a subsequent paper).

B) Simulation Results

The time domain graphs all show a partial voltage collapse and large over-currents caused by the single phase to ground fault. The circuit breaker responded between one and three cycles later and causes a total voltage and current collapse. Following the breaker operation the voltage attenuated over a short period as the isolated trapped charge is reflected up and down the line. The frequency spectra were obtained using MATLAB's inbuilt FFT routine.

It should be noted that the circuit breaker response varies from one to three cycles, due to the breaker logic requiring a coincidence of a voltage zero followed by a current zero. However, since the frequency domain information is of interest, the inconsistent response time of the circuit breaker does not significantly affect the results.

The study showed that there was significant penetration of harmonics onto the adjacent lines. This can be seen in Fig. 4, which shows the frequency spectrum for the healthy phase

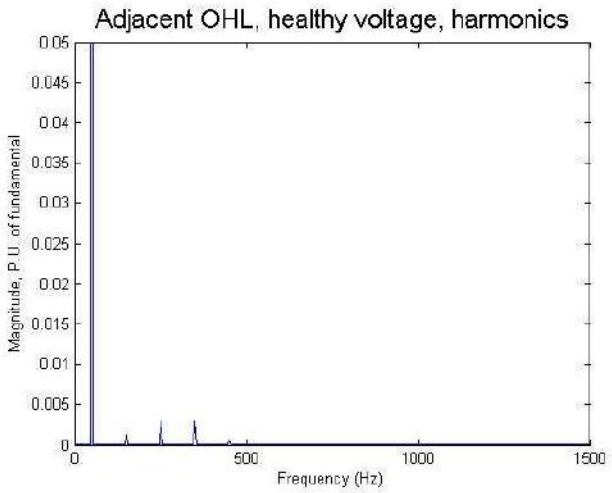


Fig. 4. Voltage frequency spectra, healthy phase, adjacent line with harmonics.

voltage for the sending end bus bar. This can be directly compared to the control case shown in Fig. 5 that did not have harmonic sources at Farr and Paul's Hill wind farms.

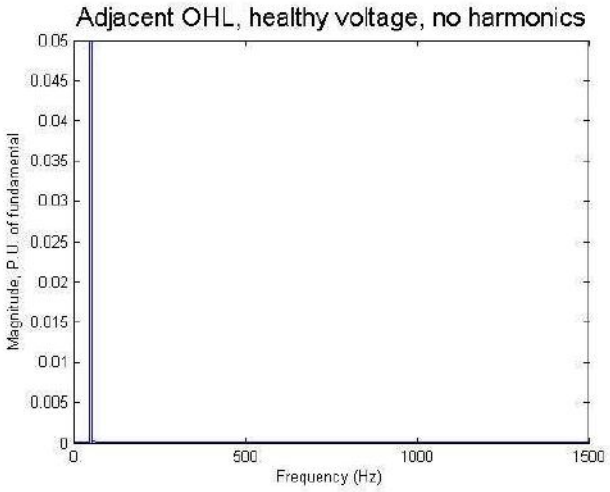


Fig. 5. Voltage frequency spectra, healthy phase, adjacent line without harmonics

In the case of the adjacent line, there is significant generation nearby, and so the transients were suppressed due to the local strength of the grid. This can be seen in Fig. 6 and Fig. 7, which show the voltage time domain for the sending end with harmonics and the control case respectively. The effect of mutual coupling in the healthy phases was particularly suppressed: see Fig. 5, Fig 6 and Fig 7.

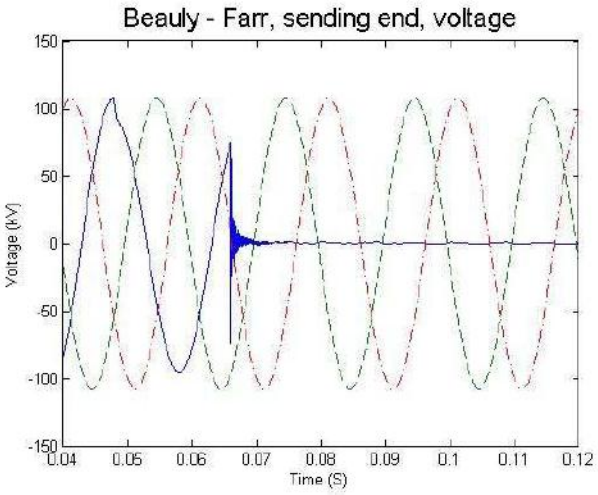


Fig. 6. Time domain voltage, all phases, adjacent line with harmonics.

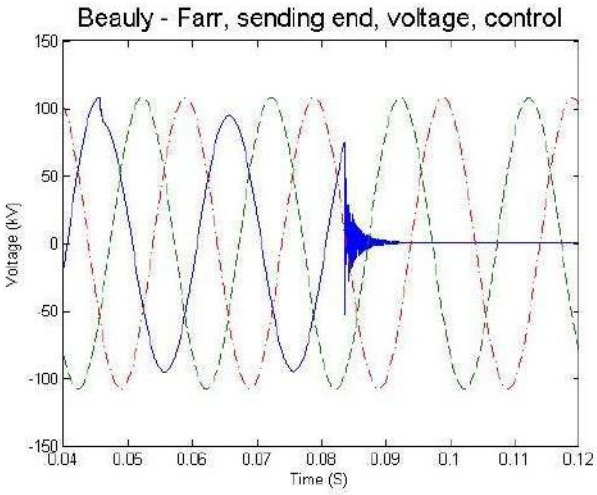


Fig. 7. Time domain of voltage, all phases, adjacent line without harmonics.

In comparison, the distant line has significant mutual coupling between the healthy and faulted phases on the voltage waveforms. This can be observed in Fig. 8, which shows the frequency spectrum for one of the healthy phases *without* harmonic sources.

Due to this pronounced mutual coupling, it cannot be determined to what extent the harmonics penetrate onto the voltage signals at this location on the grid. This is so because the coupling swamps the frequency spectrum on the distant line. The magnitude of the coupling here is even greater than the harmonics on the adjacent line. Comparison can be made between Fig 4 and Fig 8. In other words, if coupling to this extent was present on the adjacent line, the harmonics would not be clearly observed even though this line is electrically nearer the wind farms.

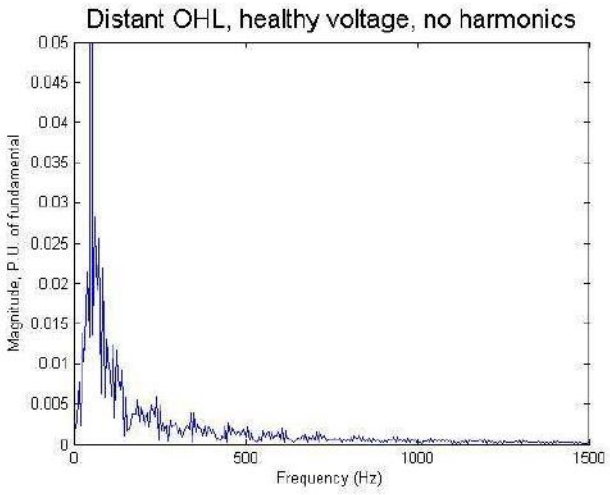


Fig. 8. Voltage frequency spectra, healthy phase, distant line without harmonics

However, the current signals and corresponding spectra of the distant line (Fig. 9 and Fig. 10) suggest that wind farm harmonics do not significantly penetrate this far. This can be concluded when these plots are compared to healthy phase current frequency spectra on the adjacent line with harmonics, shown in Fig. 11. Incidentally, since the absolute current on the healthy phases was much smaller in comparison, there is very significant harmonic component in proportion to the fundamental. The reduced current is due to much less real power transfer on both circuits between Farr and Beauly than on those between Elgin and Nairn.

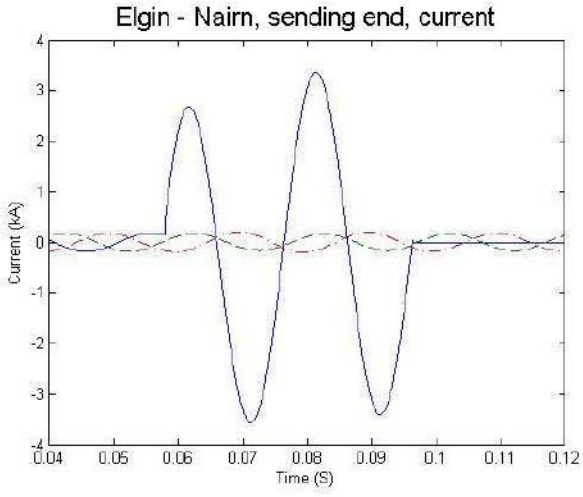


Fig. 9. Time domain of currents, all phases, distant line without harmonics

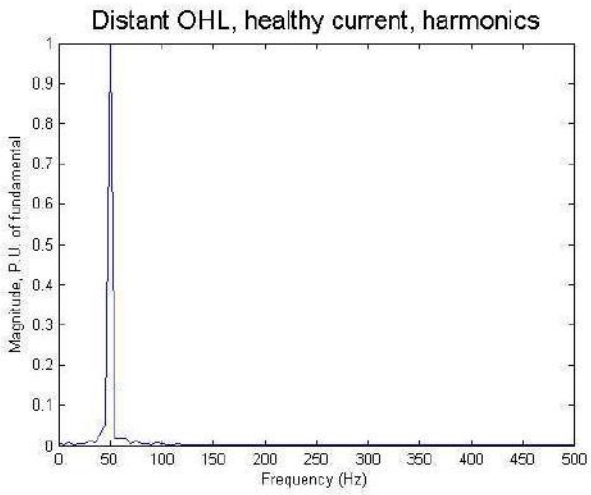


Fig. 10. Current frequency spectra, healthy phase, distant line with harmonics

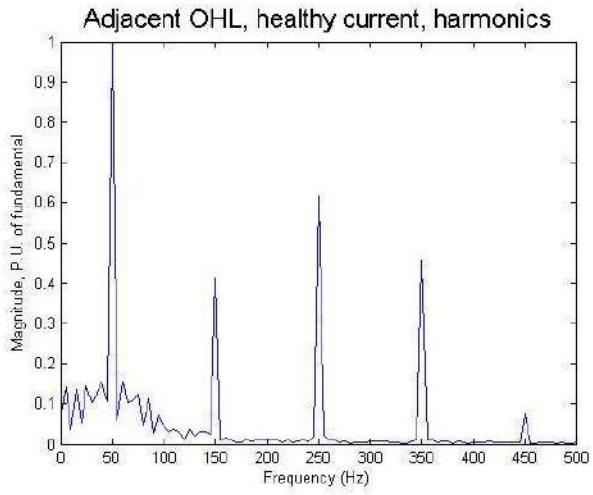


Fig. 11. Current frequency spectra, healthy phase, adjacent line with harmonics

C) Discussion

Typical AA schemes rely on diagnosing the extinguishing of the secondary arc using a combination of signal processing and AI, such as neural networks or fuzzy logic [3]. The time series information is transferred into the frequency domain, and different bandwidths form the inputs to an AI inference system. The AI is able to discern non-linearity between the input and output data and able to generalize in the presence of unknown fault cases.

The harmonic study shows the power system topology is more important than the presence of harmonics, particularly on the distant line. Since AA schemes using AI can overcome varying primary system parameters, this suggests they will be robust in the presence of wind farm harmonics. However, on nearby lines it would be prudent to include harmonics signatures in the calibration of adaptive aurtoreclosing relays.

Design of adaptive autoreclosure schemes for 132kV with high penetration of wind: Part 2 – Real time development and testing

Le Blond S. P., *Student Member, IEEE* Aggarwal, R., *Member, IEEE* and Ross B.,

***Abstract* — Following from part 1, this paper describes the development and real time testing of an adaptive autoreclosing scheme for a 132kV system with high penetration of wind generation. Having previously established the real time system model, this second part concentrates on the hardware and software tools used to build the relay development platform, the algorithm itself, and the testing of this algorithm. A results section documents test of the neural network-based algorithm and is shown to be 100% reliable. Further successful tests on a 400kV system suggest the algorithm is robust to other system configurations.**

***Index Terms* — Adaptive reclosing, transmission lines, power system protection, real time simulation, wind generation.**

I. Introduction

FOR the purposes of this paper, adaptive autoreclosing (AA) on overhead lines is the ability to distinguish between a transient and permanent fault following a short circuit, and then issue a reclose signal if and only if the transient fault no longer exists. This approach is beneficial for quickly restoring supply, increasing system stability and minimising secondary shocks to the system due to unsuccessful reclose attempts. Single phase adaptive autoreclosure takes advantage of around 80-90% overhead line faults being transient and 70-90% involving a single phase to ground [1]. It therefore demands correct phase selection and the ability of the circuit breaker to open a single phase, and leave the healthy phases intact. A permanent fault has a constant fault resistance since it usually involves a physical short circuit, due to vegetation, downed line or broken conductor. In contrast, a transient fault involves arcing across the arcing horns and is usually caused by lightning or adverse weather conditions. The heavy current primary arc before the circuit breaker opens is difficult to detect because although the resistance is dynamic, it stays in the order of a few ohms, similar to the permanent fault. Following the single-phase breaker opening, the healthy phases, (those that remain energized) mutually couple and drive a highly non-linear, lower current secondary arc on the faulted phase. When this arc eventually extinguishes it is safe to reclose the circuit breaker and bring the line back into normal service. The notable difference between secondary arcing and a permanent fault gives rise to the possibility of robust diagnosis between the two cases. However there is a complex interplay of parameters that determine both voltage signatures, some of which cannot be known pre-fault. It is therefore problematic to rely solely on power frequency measurements. In the past, researchers have addressed this problem a number of ways, including signal processing and neural networks [2], fuzzy logic and wavelet transforms [3] and straight forward numerical techniques [4]. These methods all rely on the higher frequencies caused by the secondary arc. However, only the artificial neural network technique has been deployed on a real system *and* documented in the literature [5] and therefore forms the basis of this algorithm.

In recent years, the UK has seen a considerable increase in wind power in the generation mix, and around half of this installed capacity is currently onshore [6]. Modern variable speed turbines involve power electronics in the AC/AC conversion process to export power at the desired grid frequency. This is usually in the form of a back-to-back converter whose individual gates are pulse-width modulated. The result of this approach is the injection of harmonics onto the local grid. Filters are therefore required to eliminate lower order harmonics such that utility grid codes are met. One concern about wind farm power quality is its effect on transient based protection, such as AA, due to its reliance on higher frequencies.

The primary purpose of this paper is to assert the conclusions drawn in [7], that in fact, AI autoreclosing techniques are robust to the presence of wind farms. This is achieved by demonstrating an ANN based AA scheme in real time on a transmission line adjacent to a wind farm. The real time platform used is described in the earlier paper [8]. This paper concentrates on the real time hardware, software and the algorithm itself.

Seldom in the literature are novel protection algorithms demonstrated in real-time, closed loop control. The secondary purpose of this paper is to describe a low cost, versatile relay development platform achievable within an academic environment. It is hoped that such an approach may greatly hasten commercialization of research output in power system protection.

Manuscript received March 9, 2011. This work was supported by Supergen: Flexnet, an EPSRC sponsored research consortium.

S. P. Le Blond, R. K. Aggarwal and B. Ross are with the department of Electronic and Electrical Engineering department, University of Bath. (email: s.p.leblond@bath.ac.uk)

II. Problem Space

The task of adaptive autoreclosure is, in essence, one of pattern recognition. Fig. 1 a) shows a transient fault and fig. 2 b) shows a permanent fault time series for a 400kV system generated by ATP draw. These are taken from the line side of the breaker on the faulted phase, being essentially what the relay 'sees', albeit via the CVT, or other transducer, in order to transform to practical secondary system voltages.

At point 1 on fig. 1a) the transient arcing fault occurs. The protection signals the circuit breaker operation at point 2 and the secondary arc begins, which eventually extinguishes, leaving a plain permanent sinusoid. On fig. 1 b), the same sequence of events occurs, except the resistance of the fault is fixed. After point 2 the circuit breaker operates, and after a short period of transients a bare permanent sinusoid remains.

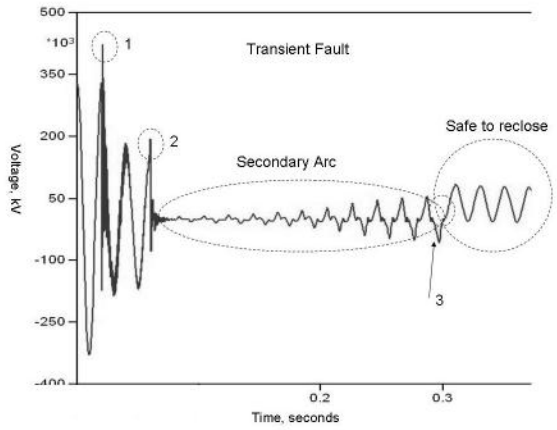


Fig. 1a) Faulted phase voltage time series of a transient fault.

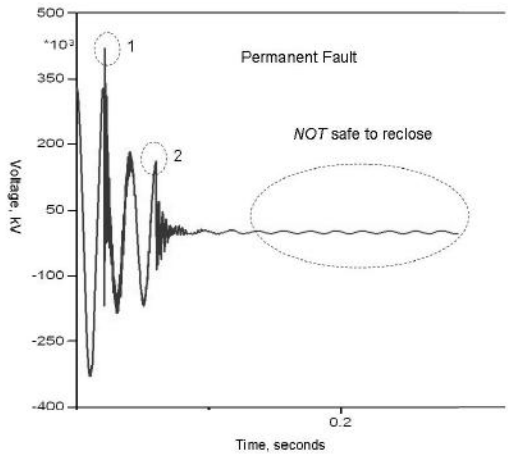


Fig. 1b) Faulted phase time series of a permanent fault.
These figures show the cases adaptive autoreclosing relay must differentiate between.

In terms of the autoreclosing problem there are two important conclusions to draw from figs. 1 a) and b). The voltage trace between 1 and 2 show very little variation and thus renders diagnosis between fault types difficult in this period, discussed further in section II. In addition, the post arcing period, on fig. 1 a) and the permanent fault, fig. 1 b) are both plain sinusoids but belong to different classes in the problem space. In the technique described by Fitton et al. [2], a single ANN is used to diagnose between *safe to reclose* or *not safe to reclose* conditions with respect to the time series. An important innovation on the method is to assign two neural networks, A and B, separate tasks. A is assigned the task of arc fault type determination and B is assigned the task of arc extinction. This means that reclosure is sanctioned by the arc extinguish network, only if the type detecting network deems the fault to actually be transient. This increases the robustness of the method because the networks are experts at their particular task.

III. Arc Model

The arc model is possibly the most important aspect of the simulation in the 132kV system. This is necessary to model the dynamic fault resistance associated with the transient arcing fault. The arc conductance is described by (1), the equation for unconstrained arcs in air.

$$\frac{dg}{dt} = \frac{1}{\tau}(G - g) \quad (1)$$

Where in (1) G is stationary arc conductance, g is time dependent arc conductance and τ is the time constant. These parameters are themselves dependent on various others that are empirically determined by volt-current cyclograms, [9] and [10]. The behavior of the arc occurs in two distinct stages, before and after the circuit breakers open. The primary arc is a heavy current, high-energy arc, fed by the short circuit on the associated phase conductor. In this stage, the arc length does not evolve appreciably with time and the average resistance is in the order of a few ohms, similar to a permanent fault resistance. This similarity to a permanent fault renders the phenomena extremely difficult to diagnose with conventional transducers. A technique has been proposed [11], but it relies on line traps to detect very high frequencies. The RTDS system guarantees fidelity of up to 3kHz, but the frequency response of most conventional CVTs is only faithful to about 700Hz [12]. The frequency response begins to attenuate after this since the CVT equivalent circuit acts like as low pass filter. It is therefore only realistic in this context of real time simulation to make a robust diagnosis using the secondary arcing period. The graph shows the similarity in the frequency domain between the permanent fault and the primary arc up to 2kHz, showing why it is not feasible to use frequency domain information for such purposes.

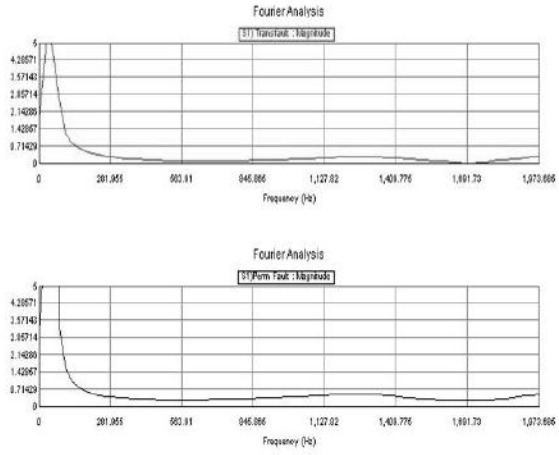


Fig. 2. Frequency domain plots of the primary arcing period against the permanent fault from 0-2kHz. The similarity makes pre-circuit breaker fault diagnosis hard to achieve.

RTDS include a precompiled fault arc model in their draft library of the RSCAD software. A detailed explanation of how the arc model is achieved is available in [13] but briefly, integrator control blocks solve equation (1) for arc conductance on a per time-step basis. Evolution of the secondary arc length is controlled by random white noise and Gaussian noise generators. This helps keep a realistic element of randomness to the arc signature and means that no two test runs are the identical. Reference [13] shows the model has good agreement with real world fault tests. In this work, a permanent fault is represented by a permanent resistance of 2ohms to ground.

IV. The Algorithm

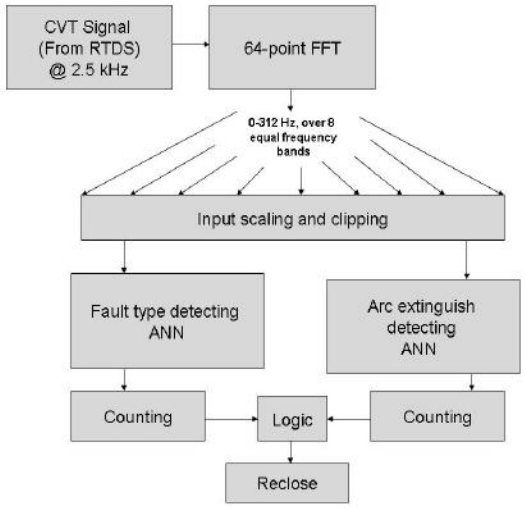


Fig. 3 A simplified block diagram showing how the CVT signal is processed to arrive at a trip decision. An arc must be detected *and* extinguished to arrive at a trip decision.

The data acquisition module, the X3-SD, has an oversampling facility and internal anti-aliasing filters. This sample rate is set to 20 kHz and the input sample rate from the RTDS is then down sampled to 2.5 kHz, forming the input time series to algorithm shown in fig.3. The down sampling makes the size of the frequency transform per timestep much less computationally onerous. The time domain information is then fed to a 64-sample buffer. The buffer is then Hann windowed and the resultant subjected to the fast fourier transform (FFT). The purpose of the Hann window is to minimise edge effects and weight the window such that the frequencies in the centre have the maximum influence. The frequency bands are 39.1Hz wide. This captures the power frequency component as well as giving the best FFT frequency resolution. Extensive simulations show that in the secondary arcing period on the transient fault, only the first eight frequency bands show significant information, up to around 300 Hz. Fig. 4 shows a 3 dimensional spectrogram *looking back across the time axis.* (This perspective makes the secondary arcing frequencies easier to observe). Only the first eight frequency bands are used and the rest are discarded. The time-varying magnitudes over these eight frequency bands form the input layer of two multilayer perceptrons. However, the frequency boundaries must undergo an important intermediate stage of input scaling and clipping. This serves to normalize all inputs to the multilayer perceptron between 1 and -1. The upper and lower bounds to each frequency band, X_{max} and X_{min}, are determined offline by secondary arc training waveforms. The transfer function may be summarized by equation 2.

$$Y_{out} = f(x) = x\left[\frac{2}{X_{\max} - X_{\min}}\right], \quad X_{min} < x < X_{max}$$

$$Y_{out} = 1 \quad , \quad x > X_{max}$$

$$Y_{out} = -1 \quad , \quad x < X_{max} \qquad (2)$$

V. Multilayer Perceptron

The computing tool at the heart of the algorithm is an artificial neural network (ANN), specifically, of multilayer perceptron (MLP) architecture. MLPs are feed forward networks, trained using supervised learning techniques. They are capable of discerning non-linearity, can recognize trends in data and are robust to noise. They use supervised learning algorithms to adjust their synaptic weights so that they behave in the desired manner for the application. The ANNs are fed 8 input frequency bands from the discrete time series sampled from the secondary arc voltage. The input is then routed through the hidden layer onto a single output neuron that indicates the Boolean diagnostic task it has been trained for.

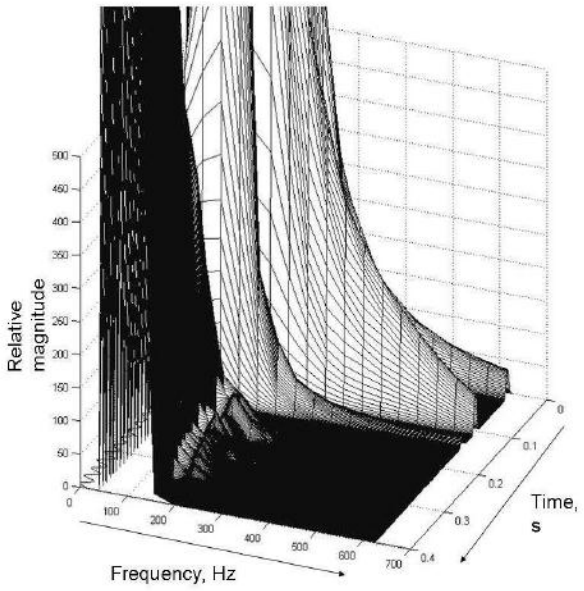


Fig. 4. A 3D spectrogram showing how the faulted phase voltage varies in the frequency and time domain, looking back across the time axis. The front axis shows frequency. The secondary arc begins at about 0.15 seconds, and only the frequencies up to 300Hz are particularly significant thereafter.

The algorithm developed uses two ANNs. One network diagnoses the presence of the arc whilst the other diagnoses the arc extinguish. The networks were developed in MATLAB's Neural Networks toolbox. Both networks are comprised of an 8-12-1 architecture, that is, 12 hidden layer neurons. Various architectures were tried, but most trial ANNs performed equally well. Within wide limits, the size of the hidden layer did not significantly influence the network's ability to perform their designated task in this application. The robustness is due to a well-defined problem space in both cases: the fault type and extinguish are both easily characterized by an abundance, or lack thereof, of higher frequency components in the 50 - 300 Hz bands. The transfer functions in the hidden layer and the output layer were selected to be logistic sigmoid and the training algorithm was matlab's resilient back propagation. The output neuron is trained to output a zero or one indicating in which class the input time series belongs. This method is adapted from [13]. In practice, output of each ANN is rarely precisely 1 or 0, however, so output above a threshold of 0.5 is rounded to indicate 1 and below 0.5, to indicate 0. Clearly, each decision cannot be based on a single windowed FFT, as this would not be particularly robust. The network does occasionally swing into the wrong domain due to frequency anomalies. In the case of the arc extinguish ANN, this is compounded by a dither region when the secondary arc extinguishes. Over this period the windowed FFT takes both information form the *not safe to reclose* and the *safe to reclose* periods. This is avoided in the ANN training cases, but cannot be avoided when the algorithm is deployed online since no foreknowledge exists of when the arc extinguish will occur. To increase robustness, a counting mechanism is introduced such that both ANNs must have a consistent output of above 0.5 for a full power system cycle before a decision is reached. In the case of the fault type ANN, the decision is framed such that the counter must register an arcing fault or the fault is assumed to be permanent, thus minimising the chance of reclosing onto an arcing fault. In practice, the marked difference between the arcing fault and the permanent fault means that the fault is always correctly diagnosed well before any existing arc extinguishes.

Fig. 5a shows a typical time domain response of the ANNs to a transient fault and fig. 5b response to a permanent fault. The middle trace on both figs. show the arc extinguish detecting ANN and the last trace the type detecting ANN. In the case of 5b, the second trace is ignored by the algorithm since the fault is correctly diagnosed as permanent. In other words, there is no arc extinguish to detect.

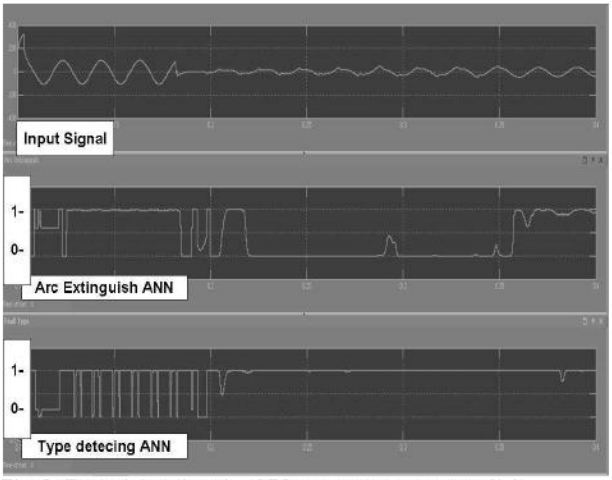


Fig. 5a Typical time domain ANN response to a transient fault.

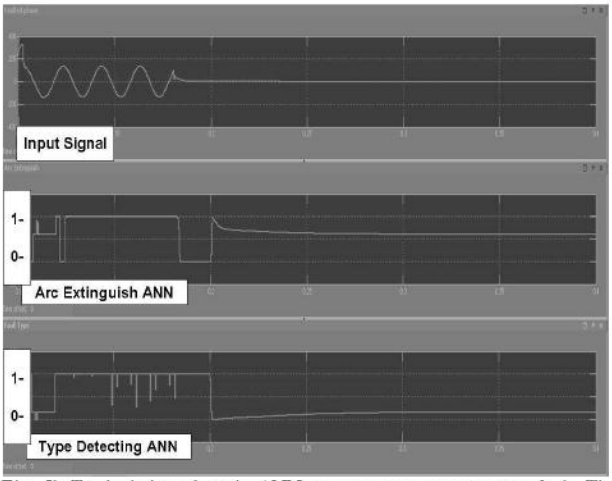


Fig. 5b Typical time domain ANN responses to a permanent fault. The middle trace in this case is not used since the fault is correctly diagnosed as permanent.

VI. Pseudo Relay Development and Test Platform

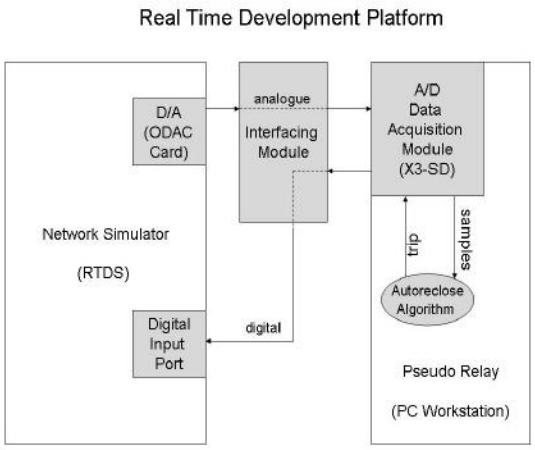


Fig. 5. Block diagram of the real time development platform

The purpose of the pseudo-relay platform is to provide an authentic system that can be decoupled from the RTDS, receiving analogue system inputs in much the same way a substation relay would from a CVT. The possibility of digital in also makes the platform potentially capable of IEC 61850 compliant interfacing. The on board data acquisition module with built in FPGA turns the PC into a versatile prototyping station with considerable processing power. Using this approach, any problems in the development stage such as prohibitive computational overheads or robustness are overcome making commercialization more attractive to a necessarily conservative industry. Fig. 5 is a block diagram showing the real time development platform as well as indicating how the pseudo relay system connects to the RTDS. The data acquisition module is innovative integration's X3-SD, a 16 channel, hi fidelity analogue capture board with an onboard 1M gate FPGA. This streams samples in packets to the workstation via PCI express bus. Sample rates of up to 216 kHz per channel are possible and streaming data rates to the host of up to 100MB/s are achievable. The purpose of the interfacing module is to convert analogue signals from the RTDS ODAC out to acceptable input levels for the X3-SD A/D module. The interfacing module also brings digital out from the A/D up to acceptable levels. A circuit diagram of the X3 SD interface is shown in fig. 6.

With reference to fig. 6, the RTDS ODAC cards supply a single ended AC signal with a peak voltage of +/- 10V. The X3-SD operates best on a differential signal, where each differential pair must be +/- 5V peak. Such an arrangement rejects any common mode noise, producing a cleaner signal. It is therefore necessary to use a differential driver coupled with a potential divider circuit in the arrangement shown in fig. 6. In the reverse direction, the reclose signal is sent to the X3-SD with digital logic. The digital out of the X3-SD runs on 3.3V logic whereas the digital input ports of the RTDS operate on 5V logic. The digital input port of the RTDS are inverted, such that 0V indicates a logical 1. A transistor is therefore wired as a switch, such that 3.3V supplied to the base will result in 0V between the emitter and collector. The Zener diode ensures that the input signal to the transistor can never exceed much more than 3.3V.

The development platform is equipped for 12 channels of analogue in and four channels of digital out. In theory, all voltages and currents from a three phase double circuit line may be used for inputs to any prototype relay, or for that matter, 12 signals from anywhere in the primary system. It should be stated explicitly however, that this application only uses 2 channels of analogue in, namely the faulted phase and circuit breaker status, and a single channel of digital out to indicate a reclose event. Following the circuit breaker operation, and consequent initiation of the secondary arc, the second channel of analogue triggers the algorithm. This approach ensures that the ANNs only act on the secondary arc and not the steady state signal, as they have not been trained to deal with the latter.

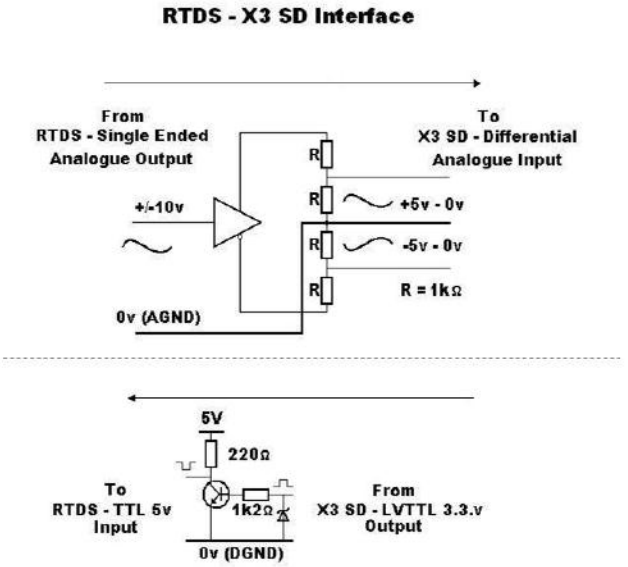


Fig. 6. A single channel circuit diagram of the interface box. The first circuit is an analogue channel to the pseudo and the second is a digital from the pseudo relay to the RTDS. The +/- 12 V supply rails are omitted for clarity.

VII. Software

The X3-SD is supplied with an extensive C++ class library known as *Malibu*. The *Malibu* library allows the user to develop custom applications to process the data as it is captured and streamed to the host. An example program, *SNAP*, that configures, arms and executes data capture is also bundled as a *Microsoft Visual Studio* project, along with support for various other development environments. The user may augment the source code with Malibu and recompile the

program to suit his or her specifications.

The algorithm was developed with Mathwork's ubiquitous engineering software *MATLAB*. In MATLAB, SIMULINK's embedded real time encoder was used to code block models into C++. This approach considerably speeds up development time as SIMULINK models can be directly coded for prototype targets, tested, and then quickly adjusted if required.

The Simulink-generated source code was imported to the SNAP project as a stand-alone method that could be invoked from the main program for every algorithm time-step. Each time a sample is received from the RTDS, this method is called, with the sample's value as the input to the method. As mentioned in section IV, it is necessary to down-sample the input to make the FFT yield the required frequency resolution. This downsampling is achieved within the algorithm, with the SNAP sample rate is set to 20 kHz, and the algorithm's code discarding seven samples in every eight. The method was set to return the relevant circuit breaker status and linked to the Malibu digital out class function, such that a 1 was sent to the digital out channel when the algorithm computed a reclose to be necessary. This signal is sent to the RTDS via the interface module.

VIII. Training and Testing

Previous studies [14] have shown that the post fault transient voltage is responsive to the following parameters: system voltage, line length, geometric line configuration, bus short circuit capacities, fault point on line, fault inception point on the waveform and permanent fault resistance. Of these, the last three parameters are not known prefault and so cannot be configured before the relay is deployed on any real world system. These parameters were therefore varied over the test cases to test the algorithm for its sensitivity to the same.

Any algorithm using ANNs must involve training the ANNs to respond appropriately to their input data. This is most effectively achieved using fault waveforms generated by varying each relevant parameter over the full range of values the scheme will encounter in practice. In a real world fault scenario, each of these parameters may take any value on a continuous range. However, in producing the training set, it is only practical to vary each parameter in large, discrete steps. This is because a large number of parameter permutations quickly combine to produce a prohibitively large training set. The RTDS model provided both training and test cases, but there were important differences in the model used for training and the model used for testing. This was to ensure that the ANNs could generalise rather than simply memorise. In other words, to ensure they were able to recognise trends in test data rather than just reproduce their performance. Moreover, the integral reason for establishing the realistic primary system model described in part 1 [] was to determine the algorithm's sensitivity to the presence of wind generation. Therefore, parameters to generate the training and test cases were varied as shown in table 1, with the training cases denoted by the first column and A test cases the second column. B Test cases used a different primary system model that will be discussed in next section.

It should be mentioned that the initial fault inception angle had no significant bearing on the secondary arc other than determining where it began, and only because the response time of the breaker was fixed by the simulation. However, the algorithm ignored the first cycle after the circuit breaker operation to allow the circuit breaker transients to attenuate. For this reason, the range of fault inception angles was not changed between training and test cases.

TABLE I
Training and Testing

Parameter	Training cases	A Test cases	B Test cases
System Voltage	132 kV	132 kV	400 kV
Bus Short Circuit Capacities	Fixed	Fixed	Fixed
Line Length	30 km	30 km	50 km
Geometric Line	Fixed	Fixed	Fixed
Fault point on line % line length from relay	100% 50% and 0%	0, 16.6%, 33%, 66% and 83.40%	2km intervals transient fault and 10km intervals permanent fault
Inception angle on waveform	90°, 30°, 0°.	90°, 30°, 0°.	90°, 30°, 0°.
Perm fault resistance	0, 2, 50, 100 Ω	0.00001,1, 35, 80 Ω	0.00001, 35, 80 Ω
Wind Farms at Farr and Paul's Hill	No, voltage source behind equivalent subtransient reactance	Yes	N/A

Table I shows how parameters were varied to generate the training and test cases

When staging fault tests, it is necessary to split the line model into two sections with a 'virtual bus' in the centre, where the fault may be grounded to simulate a single phase to ground fault. Unless the line is split equally, splitting a 30 km transmission line will result in a line section below 15 km. In the test sets, such sections must be represented with a simple PI model, rather than a travelling wave model, because the travel time is below the fixed RTDS timestep of 50μs. However, the good performance of the algorithm with both model types suggests it is not sensitive to this approximation.

Due to the inherent randomness in the arc model, specifically evolution of the secondary arc length, no two

transient fault test runs are exactly identical. In practice, other parameters are more significant and the secondary arc extinguish is usually within one cycle of another run with the same settings.

IX. Real Time Testing

Fig. 7 shows a typical real time reclose sequence, as observed by the response of the faulted phase CVT voltage. The breaker recloses about 2.5 cycles following secondary arc extinction, and the system is restored to its nominal voltage.

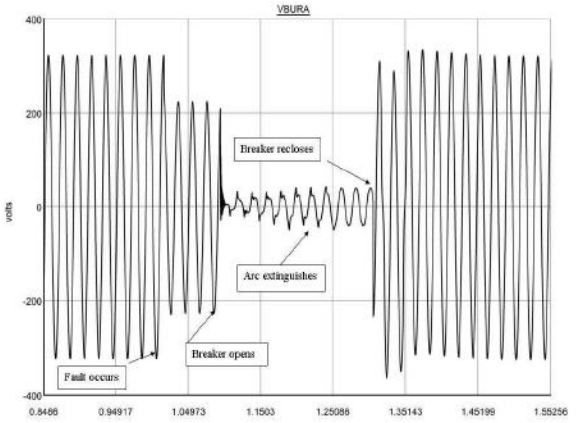


Fig. 7. A typical real time response to a transient fault, the faulted phase voltage viewed via the CVT output.

Conversely, fig. 8 shows a typical response to a permanent fault. In this case, autoreclosure is not sanctioned and voltage is not restored as the fault is correctly diagnosed to be permanent.

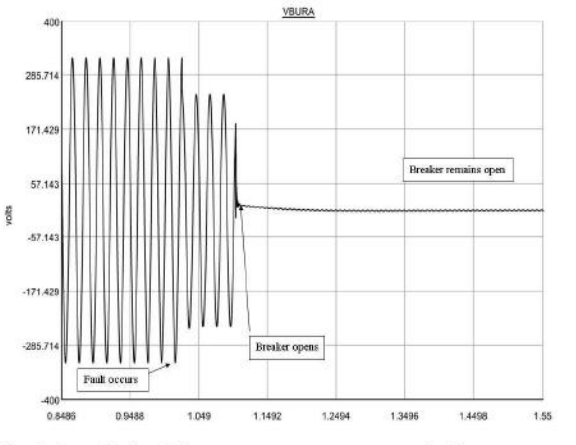


Fig. 8. A typical real time response to a permanent fault.

The benefit to the system is best viewed through the short circuit currents sustained on the transmission line. These can be seen in fig. 9 and 10. With reference to fig. 9, when the fault occurs there is a large p.u. short circuit current on the faulted phase. This then collapses to zero when the circuit breaker isolates this phase. Following the reclose event, a healthy short circuit current is re-established.

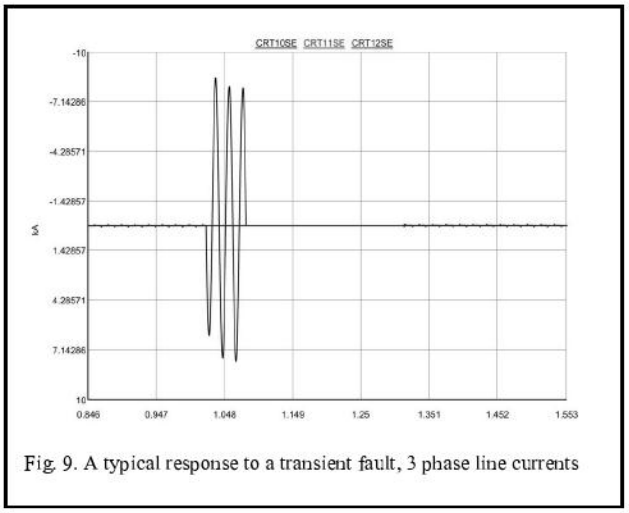


Fig. 9. A typical response to a transient fault, 3 phase line currents

Fig. 10 shows the line currents when the autoreclosing algorithm is bypassed and reclosing is forced before arc extinction. In this case, the primary arc re-ignites and large fault currents recur. A similar sequence of events occurs in the case of reclosing onto a permanent fault. Although in practice, the protection would soon retrip the breaker following an erroneous reclose, sustaining a second fault is nonetheless hugely detrimental to the system. Ohmic heat is unlikely to dissipate in this time and therefore the fault damage is likely to be at least doubled. Comparison of the line currents between fig. 9 and 10 directly shows the benefit of intelligent autoreclosing.

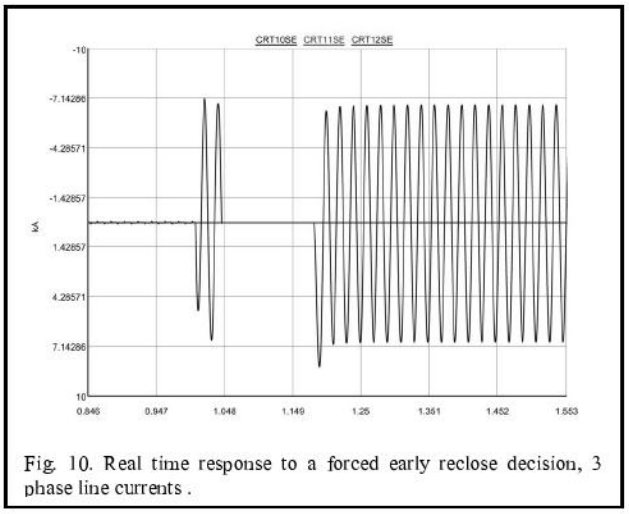


Fig. 10. Real time response to a forced early reclose decision, 3 phase line currents .

In some cases, fast autoreclosing has been shown to enhance transient stability. However, it should be noted that transient stability is not a concern in this system, especially when considering single phase to ground faults. In the first instance, power transfer through a double circuit, 132kV 30 km line, in an interconnected system is limited by thermal limits rather

than transient stability limits. Study of the rotor angle for a single phase to ground fault also shows that the system can sustain this fault condition indefinitely without losing stability.

With reference to table 1, the tests on test cases A were 100% successful, in that for every transient test case, the algorithm recognized the extinction of the secondary arc and safely reclosed, and for every permanent case, autoreclosure was blocked. It can therefore be concluded that this algorithm is robust to the presence of wind farms, at least in the context of this system model.

Some time after these tests were completed, the algorithm was tested on a different primary system, the parameters varied as per table I column 3. This was a 50 km, 400kV twin circuit line typical of the UK transmission system. The wind farm models were not present in these test cases. The main difference in this system was increased secondary arcing energy due to the increased system voltage. With no changes, the algorithm performed poorly. This was at first attributed to poor ANN response, the assumption being they required retraining for a different primary system. However, further investigation revealed poor performance was due to saturation of input levels on the hard limiter, described by equation (2) section IV, confusing network output. When the input level from the CVT signal was reduced to 50% of that of the 132 kV input level, the algorithm was 100% successful in all 140 tests. This indicates that the input scaling and clipping stage is as important as the network itself and an important area for further work.

IX. Conclusion

An accurate, real world primary system model has been built, together with a versatile relay development platform. Using this platform, an adaptive autoreclosing algorithm has been developed and demonstrated in real time. Extensive tests reveal that the algorithm is 100% successful in the context of this system and is not sensitive to DFIG based wind generation. A second test suggests that, subject to proper input scaling, the algorithm is robust to other transmission systems. However further work on whole range of system configurations, combined with real world fault data is required to confirm this last conclusion.

References

[1] Areva T&D, "The Network Protection and Automation Guide", 2008, Available at www.alstom.com/grid/NPAG

[2] D. S. Fitton, R.W. Dunn, R.K. Aggarwal, A.T. Johns and A. Bennett, "Design and implementation of an adaptive single pole autoreclosure technique for transmission lines using artificial neural networks", IEEE Transactions on Power Delivery, 1996, vol. 11, p. 748-756

[3] H. Khorashadi-Zadeh and Z. Li., "Transmission Line Single Phase Auto Reclosing Scheme Based on Wavelet Transform and Adaptive Fuzzy Neuro Inference System", in 39th North American Power Symposium, Las Cruces, NM, 2007, p. 43-48.

[4] V. V. Terzija and Z. M. Radojevic, "Numerical Algorithm for Adaptive Autoreclosure and Protection of Medium-Voltage Overhead Lines", IEEE Transactions on Power Delivery, 2004, vol. 19, p. 5.

[5] I. P. Gardiner and J. Ramsden, "On Site Experience of an Adaptive Autoreclose Relay for ovehead lines", IEE conference publication on developments in power system protection, Nottingham,1997, p. 377-380.

[6] BWEA, *British Wind Energy Association Official Website*, 2008, [cited 2008], Available at: http://www.bwea.com.

[7] S. P. Le Blond and R. K. Aggarwal, "Impact of Wind Farms on Electro-Magnetic Transients on 132kV Network with Particular Reference to Fault Detection" CIRED, Prague, 2009.

[8] Le Blond, S. P. and Aggarwal, R. K. "Design of adaptive autoreclosure schemes for 132kV network with high penetration of wind: Part 1 - Real time modeling", Transmission and Distribution Conference and Exposition, 2010 IEEE PES.

[9] Johns, A.T., R.K. Aggarwal, and Y.H. Song, "Improved techniques for modelling fault arcs on faulted EHV transmission systems. Generation, Transmission and Distribution, IEE Proceedings", 1994, 141(2): p. 6.

[10] Song, Y.H., R.K. Aggarwal, and A.T. Johns, "Digital simulation of fault arcs on long-distance compensated transmission-systems with particular reference to adaptive autoreclosure", European Transactions on Electrical Power Engineering, 1995, vol. 5, p. 315-324.

[11] Z. Q. Bo, R.K. Aggarwal, A.T. Johns, B.H. Zhang and Z.Y. Ge. "New concept in transmission line reclosure using high frequency fault transients", IEE Proc. in Generation Transmission and Distribution, vol. 144, p. 351-356, 1997.

[12] Fitton, D., "A Neural Network Based Adaptive Single Pole Autoreclose Technique", *(PhD thesis)*, Department of Electronic and Electrical Engineering, University of Bath, Bath, 1995.

[13] Giesbrecht, J., Ouellette, D.S. and Henville, C. F., "Secondary Arc Extinction and Detection - real and simulated", DPSP, Glasgow 2008.

[14] Johns, A.T. and Aggarwal, R.K., Digital simulation of faulted EHV transmission lines with particular reference to very high speed protection, Proceedings of the IEE London, 1976 **123**(4): p. 353-359.

Acknowledgment

The authors would like to thank Paul Forsyth and his colleagues at RTDS technologies, Jim Henderson at Innovative Integration and John Owen, Kathy Ollington and Tim Bigg at Entegra DSP.

Simon P. Le Blond (M'2009) is a PhD student at the department of Electrical and Electronic Engineering, University of Bath. He attained a BSc Hons degree in Physics from the University of Southampton in 2004. His main research interests are real time power system simulation, power system transients, power system protection and AI techniques applied to power systems.

Raj Aggarwal (SM'91) attained his degrees of BEng and PhD from the Department of Electronics and Electrical Engineering, University of Liverpool, in 1970 and 1973 respectively. He then joined the University of Bath where he is now Professor of Electrical Engineering and head of the Electrical Power and Energy Systems Group in the Department of Electronic and Electrical Engineering. His main areas of research are modelling of electromagnetic transients in power systems, development of novel protection/fault location techniques for flexible transmission/distribution systems, including smart grids, HVDC networks, condition monitoring of power system plant and power quality. He has published over 400 technical papers and is the recipient of several IET premium awards for papers published. He is/has been on several IET and CIGRÉ committees and also has been a member of the EPSRC Scientific Advisory committee on energy. In 2005 he was awarded the degree of DEng by the University of Liverpool for his original and outstanding contribution to Electrical Power Systems.

Brian Ross